Multivariate Analysis in Ecology
– Lecture Notes –

Jari Oksanen[1]
Department of Biology
University of Oulu

2004

[1]This version: February 17, 2004

Contents

List of Slides

129 Constrained Correspondence Analysis (CCA)
130 CCA: Algorithm
131 CCA: Alternating regression algorithm
132 Those numbers...
133 CCA plot
134 Class constraints
135 Predicted values of constraints
136 LC or WA Scores?
137 WA and LC scores with class constraints
138 LC scores are the constraints
139 Number of constraints and the plot
140 Number of constraints and curvature
141 DECORANA in Disguise
142 Polynomial Constraints: A Bad Idea
143 Constrained horseshoe
144 Levels of environmental intervention
145 Significance of constraints
146 Permutation statistic
147 Number of permutations
148 What is permuted?
149 Selecting constraining variables
150 Automatic stepping is dangerous
151 Components of Variation
152 Negative Components of Variation
153 Comparing methods
154 Community pattern simulation
155 Short gradients: Is there a niche for PCA?
156 Long gradients: DCA or NMDS
157 Handling curves
158 Extended dissimilarities and step-across
159 Analysed using modern methods...
160 Classification
161 Classification of classification
162 Cluster Analysis
163 Clustering strategies
164 Clustering and space
165 Interpreting clusters
166 Number of clusters
167 Optimizing classification: K–means clustering
168 Fuzzy clustering
169 TWINSPAN: Two-Way Indicator Species Analysis
170 Criteria for good classes
171 Example: A real class structure...
172 ...And all methods fail
173 Classification and ordination
174 The choice of clustering method
175 Data Import to R
176 Preparing a spreadsheet
177 Comma separated values
178 Community data and Environmental data
179 Names

1 Site description

1.1 Diversity

There is a plenty of hype about diversity indices, but they are best seen as simple measures of variance of species abundances. Consequently, it does not matter so much what brand of diversity index is used.

Slide 1

Site Description

1. Diversity indices
2. Species abundance models
3. Species – area relationship

Slide 2

Shannon diversity

$$H = -\sum_{j=1}^{S} p_j \log_b p_j$$

Originally information theory with base $b = 2$: Average length in bits of code with shortest possible unique coding

- The limit reached when code length is $-\log_2 p_i$: longer codes for rare species.

Biologists use natural logarithms (base $b = \mathrm{e}$), and call it H'

Information theory makes no sense in ecology: Better to see only as a variance measure for class data.

Note on slide 2. Different logarithm bases are not really important, since it is trivial to transform between bases. If H' is a diversity calculated with base e, or natural logarithms ln, it can be transformed into base 2 diversity with $H = H'/\ln(2)$. Similar transformations apply for all bases and diversities with any base are linearly related.

Slide 3

Simpson diversity

The probability that two randomly picked individuals belong to the same species in an infinite community is $P = \sum_{i=1}^{S} p_i^2$.

Can be changed to a diversity measure (= increases with complexity):

1. Probability that two individuals belong to *different* species: $1 - P$.
2. Number of species in a community with the same probability P, but all species with equal abundances: $1/P$.

Claimed to be ecologically more meaningful than Shannon diversity, but usually very similar.

Note on slide 3. Hurlbert [43] asserts that Simpson diversity is ecologically meaningful because it represents the probability that two individuals of the same species meet. Even if this is a better anecdote than Shannon information, Simpson rather gives probability that two sampled individuals are of the same species: at least for plants, the probability of meeting is different [73].

Slide 4

Hill numbers

Common measures of diversity are special cases of Rényi entropy;

$$H_a = \frac{1}{1-a} \log \sum_{i=1}^{S} p_i^a$$

Mark Hill proposed using $N_a = \exp(H_a)$ or the "Hill number":

H_0	=	$\log(S)$	N_0	= S	Number of species
H_1	=	$-\sum_{i=1}^{S} p_i \log p_1$	N_1	= $\exp(H_1)$	exp Shannon
H_2	=	$-\log \sum_{i=1}^{S} p_i^2$	N_2	= $1/\sum_{i=1}^{S} p_i^2$	Inverse Simpson

Sensitivity to rare species decreases with increasing a: N_1 and N_2 are little influenced and nearly linearly related.

All Hill numbers in same units: "virtual species".

Note on slide 4. R package `vegan` contains a function `diversity` which computes Shannon diversity (with any base) and both variants of Simpson diversity [36]:

```
H <- diversity(varespec)
```

It is very easy to compute diversity even without this function. The following calculates Shannon diversity (with natural logarithms) and Simpson diversities for site number 5 in data `varespec`:

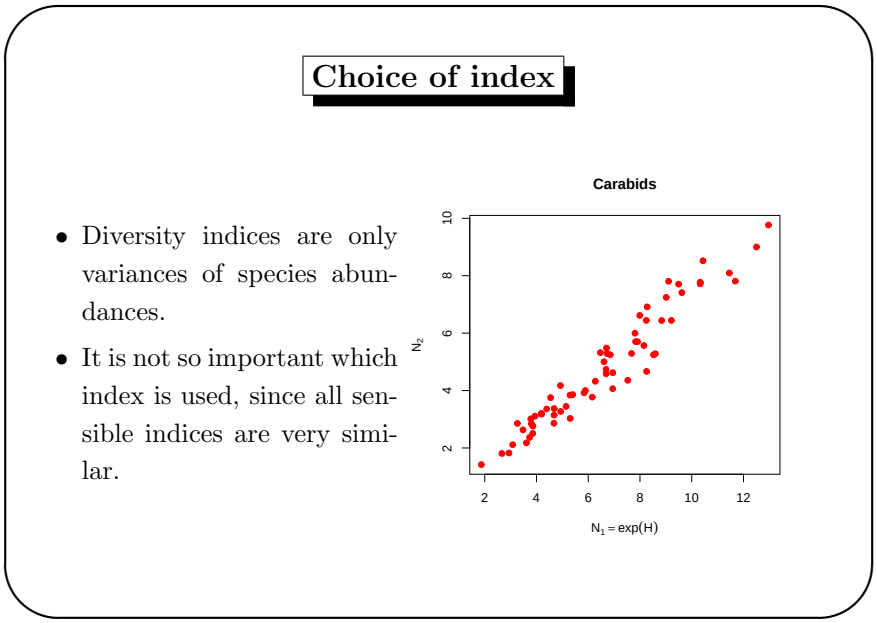
```
p <- varespec[5, ]
p <- p[p > 0]  # Remove zeros
H <- -sum(p*log(p))
P <- sum(p*p)
Simpson <- 1-P
invSimpson <- 1/P
```

The index `invSimpson` is actually Hill number N_2, and the others can be found as `N0 <- length(p)` and `N1 <- exp(H)`.

If you really want to calculate Pielou's evenness (slide 6), you can use, e.g., function `diversity`, and for the whole data set in one sweep:

```
H <- diversity(varespec)
S <- apply(varespec>0, 1, sum) ## Species richness
J <- H/log(S)
```

Slide 5



Note on slide 6. Smith and Wilson [86] analyse a large number of alternative evenness indices in addition to Pielou's. However, they share the same problems (slide 7).

1.2 Species abundance models

Diversity is actually based on species abundance models as well. However, these models deserve special treatment, since they are currrently popular with Hubbell's "Grand Unified Theory" [41]. Moreover, diversity gave only a single numeric descriptor for the variance of abundances, but now we deal with the shape of distribution of abundances.

Note on slide 8. In logarithmic series [27], the number of species f with n observed individuals is:

$$f_n = \frac{\alpha x^n}{n}$$

Here α is a diversity parameter, and x is a scaling parameter with no obvious ecological interpretation. The model is not very practical in this form, since it is difficult to fit to the data, because we should decide how to treat those abundances n with no species — and these gaps are common in the upper end of the log-series (see figure on slide 8).

Slide 6

Evenness

"If everything else remains constant", diversity increases when

1. Number of species S increases, or
2. Species abundances p_i become more equal.

Evenness: Hidden agenda to separate these two components

For a given number of species S, diversity is maximal when all probabilities $p_i = 1/S$: in Shannon index $H'_{\max} = \log(S)$

Pielou's evenness is the proportion of observed and maximal diversity

$$J' = \frac{H'}{H'_{\max}}$$

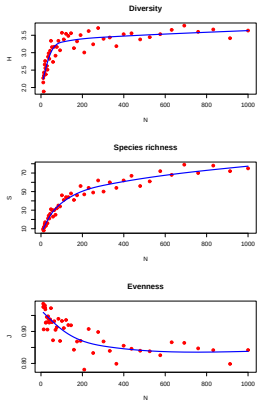
Slide 7

Sample size and diversity

With increasing sample size

- Number of species S increases
- Diversity (N_1 or N_2) stabilizes
- Evenness decreases

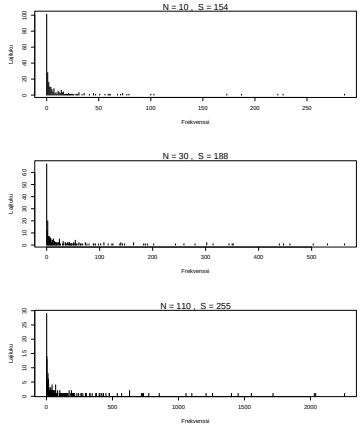
Diversity little influenced by rare species: a variance measure.
Evenness based on twisted idea.



Slide 8

Logarithmic series

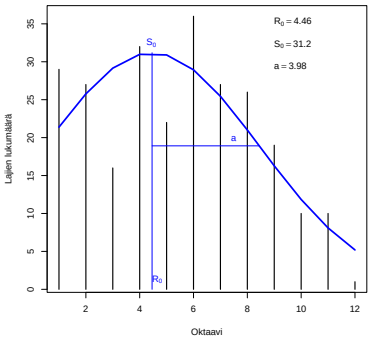
- R.A. Fisher in 1940's
- Most species are rare, and species found only once are the largest group
- In larger samples, you may find more individuals of rare species, but you find new rare species



Slide 9

Log-Normal model

- Preston did not accept Fisher's log-series, but assumed that rare species end with sampling
- Plotted number of species against 'octaves': doubling classes of abundance
- Modal class in higher octaves, and not so many rare species
- Canonical standard model of our times



Note on slide 9. Log-normal model [84] is formulated in various, and usually confusing ways in the literature. A commonly used formulation is:

$$S_R = S_0 \exp(-a^2 R^2)$$

The modal octave is designed as R_0, and the R in the equation is the difference of current octave and modal octave $R = R_i - R_0$. Species number in the current octave is S_R and in the modal octave it is S_0. The remaining parameter a describes response width.

Log-normal model is usually plotted with octaves, or doubling classes of frequencies. That is, the octaves are $1, 2, 3 \ldots 4, 5 \ldots 8, \ldots$. The example uses `carabid` data set in library `hubbell` and draws the plot both in the Fisher (slide 8) and Preston (slide 9) ways:

```
data(carabid)
freq <- apply(carabid, 2, sum)  ## Species frequencies
octave <- ceiling(log2(freq))
plot(table(freq))               ## Fisher log-series
plot(table(octave))             ## Preston log-Normal
```

We return to fitting log-Normal model later (slide 11), but we shall here have a look at the very simple method for finding the modal class and the width of the bell. It is customary to look at the class values and pick the modal class among these. However, we need not classify the data, but we can directly find the mean and standard deviation of $\log_2$ transformed data. We noted above that parameter a describes response width. However, it is not very useful directly (although customarily used), but instead we could use the standard deviation of response σ which is related to a by $\sigma = 1/(2a^2)$. We can find these directly as:

```
R0 <- mean(log2(freq))
sigma <- sd(log2(freq))
```

We can find a from σ if we really wish, but we do not, since σ is easier to interpret and can be drawn in the graph (slide 9), whereas a is arbitrary and uninteresting.

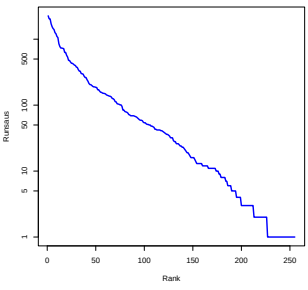
Slide 10

Ranked abundance diagrams

- Horizontal axis: ranked species
- Vertical axis: Logarithmic abundance

The shape of abundance distribution clearly visible:

- Linear: Pre-emption model
- Sigmoid: Log-normal or broken-stick



Note on slide 10. Package `hubbell` contains function `rad.lines` for drawing ranked abundance dominance models as well as for fitting some abundance models. However, it is both simple and useful for further flexibility to look how simple it is to draw such a plot in R:

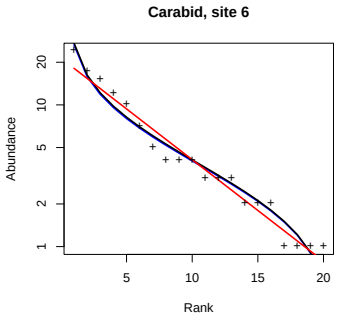
```
p <- varespec[5, ]
p <- p[p>0] # This we did already with diversity!
n <- seq(along=p)
p <- rev(sort(p)) # rev'erses ascending sort
plot(n, p, log="y",xlab="Rank", ylab="Abundance")
```

It is very common to use proportional abundances, but there is nothing in the plot nor in model fitting that requires this.

Slide 11

Fitting RAD models

- Pre-emption model
  - Species abundances decay by constant proportion.
  - A line in the ranked abundance diagram.
- Log-normal model
  - Species abundances distributed Normally
  - Sigmoid: excess of both abundant and rare species to pre-emption model.



Note on slide 11. Bastow Wilson [103] explains in detail how to fit some common abundance/diversity or ranked abundace diagram models to the data. Package `hubbell` has a (preliminary) function `rad.lines` for fitting pre-emption model and log-Normal model, and intend to port this into future versions of `vegan`.

Pre-emption model is defined by:

$$E(p_j) = \hat{p}_1(1-\alpha)^{j-1}$$

Here E means 'expected', $\hat{p}_1$ is the fitted (not observed) abundance of the most common species, and α is the proportion each species has of all remaining individuals (pre-emption coefficient). With log-transformation, this becomes a linear model:

$$\log\{E(p_j)\} = \log(\hat{p}_1) + \log(1-\alpha)\cdot(j-1),$$

with $j-1$ as the single explanatory variable, $\log(\hat{p}_1)$ as the constant, and $\log(1-\alpha)$ as the regression coefficient. The following does the job in R:

```
p <- varespec[5, ]
p <- p[p>0]
j <- seq(along=p) - 1   # Start indexing from 0
pre <- lm(log(p) ~  j)
b <- coef(pre)
hatp1 <- exp(b[1])
alpha <- 1 - exp(b[2])
```

Fitting log-Normal model is slightly more complicated. Here we assume that logarithmic abundances are distributed Normally with mean μ and standard deviation σ or formally $\log(p) \sim N(\mu, \sigma)$. In notes to slide 9 we already saw that we can find the moment estimates

of these parameters as the mean and standard deviation of log-transformed abundances. These estimates are often faily good. However, in R we have access to function `qnorm` which gives us standardized normal quantiles which we can use an explanatory variable in non-linear regression for μ and σ.

First we have to define a function that returns fitted species abundances, which are distributed Normally with mean `ln.mu` and standard deviation `sigma`:

```
> logn.fun <- function(x, mu, sigma)
{
  n <- length(x)
  sol <- exp(sigma*qnorm(ppoints(n)) + mu)
  sol[order(order(x))]
}
```

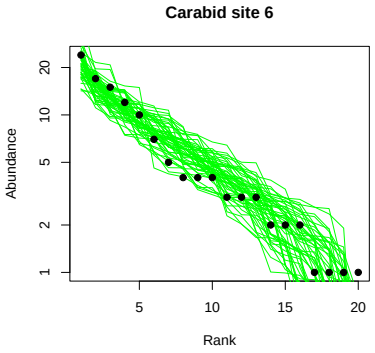
Function `ppoints` returns the evenly distributed probability points which then are used by `qnorm` for Normal quantiles. We use the moment estimates (notes on slide 9), and then fit a non-linear regression with function `nls`:

```
logmean <- mean(log(p))
logsd <- sd(log(p))
ls.fit <- nls(log(p) ~ log(logn.fun(p, ln.mu, sigma)),
              start=list(ln.mu=logmean, sigma=logsd))
```

Slide 12

Broken Stick

- Species 'break' a community ('stick') simultaneously in S pieces.
- No real hierarchy, but chips arranged in rank order:
- Result looks sigmoid, and can be fitted with log-Normal model.



Note on slide 12. Brokenstick used to be one of the most popular ranked abundance models, although it may be very difficult to know if a particular abundance distribution follows brokenstick [104]. Instead of trying to fit a brokenstick model, it may be more interesting to see if a particular abundance distribution could be a brokenstick. In R this can be done simulating a number of brokenstick models and plotting them with observed abundance distribution. This can be done with the help of function `brokenstick` in library `hubbell`. If all observed abundances are within the area of brokensticks, they are indeed a feasible model.

Note on slide 13. Hubbell's [41] model is essentially stochastic, and so it is best to simulate possible outcomes, just like in brokensticks (slide 12). For a local community, we may use Hubbell's zero-sum game. However, here we inspect a simple metacommunity

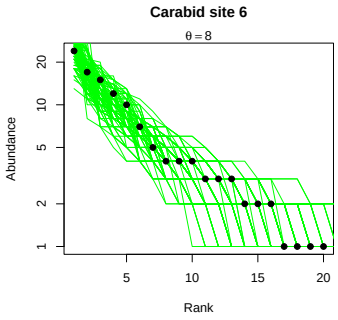
Slide 13

Hubbell's abundance model

Ultimate diversity parameter θ

- $\theta = 2J_M\nu$, where J_M is meta-community size and ν evolution speed
- θ and J define the abundance distribution
- Simulations can be used for estimating θ.

Species generator $\theta/(\theta+j-1)$ gives the probability that jth individual is a new species for the community.



model based on Hubbell's species generator, which gives the probability P that jth species belongs to a new species:

$$P(\theta, j) = \frac{\theta}{\theta + j - 1}$$

The following R function in package `hubbell` implements this species generator:

```
> hubbell.build <- function (theta, J)
{
  community <- NULL       ## Start with an empty community
  for (j in 0:(J - 1)) {
    if (runif(1) < theta/(theta + j)) ## New species?
       community <- c(community, 1)  ## Yes: add as last, abundance 1
    else {   ## No: add 1 to an old species, probabilities = frequencies
       species <- sample(length(community), 1, prob = community/j)
       community[species] <- community[species] + 1
    }
  }
  return(community)
}
```

With repeated application of this model, we may obtain a confidence envelope for feasible abundance responses with given θ and J.

1.3 Species richness

Species richness, or the number of species, is often seen as the most direct measure of biological diversity, and less problematic than diversity indices. However, it seems to be the most ambiguous measure. In particular, it is very sensitive to capture of rare species.

Note on slide 15. Rarefaction gives the expected species number of a sample of size N' that is a subset of original sample of size N, and is so an alternative to diversity indices [43]. Rarefaction sample is a subset of the total sample, and so it must be smaller than the original sample, with a maximum size $N' \leq (N - \max n_i)$, where n_i is the count for species i so that $\sum_{i=1}^{S} n_i = N$. The rarefaction expectation is given by [43]:

$$E(S|N') = \sum_{i=1}^{S} \left(1 - \frac{\binom{N-n_i}{N'}}{\binom{N}{N'}}\right)$$

Slide 14

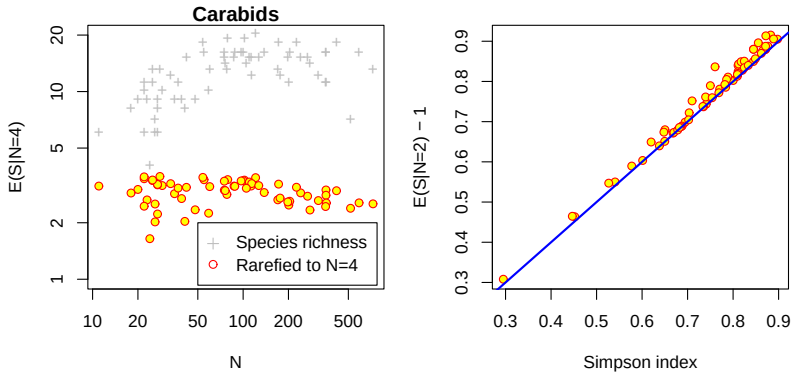
Species richness: The trouble begins

- Species richness increases with sample size: can be compared only with the same size.
- Rare species have a huge impact in species richness.
- Rarefaction: Removing the effects of varying sample size.
- Sample size must be known in individuals: Equal area does not imply equal number of individuals.
- Plants often difficult to count.

Slide 15

Rarefaction

Rarefy to a lower, equal number of individuals
Only a variant of Simpson's index



Here $\binom{N}{k}$ is a binomial coefficient, or the number of ways we can pick k items from N.

Library `vegan` has a function for rarefaction:

```
> rarefy(carabid)
Error in rarefy(carabid) : The size of 'sample' must be given --
Hint: max 4 permissible in all cases.
> S.4 <- rarefy(carabid, 4)
```

The function needs the rarefied sample size, but it will hint for the highest possible value for any data set. Rarefaction can be used only if we have real counts of individuals, and so it is usually unsuitable for vegetation data.

Rarefaction sounds simpler and more intuitive than diversity indices, but it is just another diversity index. If rarefied to $N = 2$ individuals, it will become Simpson's index plus one (slide 3) for finite samples: The first individual will certainly be a species, and the probability that the second will be a different species is given by the Simpson index, and their sum is the rarefied species richness. However, all these are adapted to finite, closed samples that I have not dealt with in these lectures since finite samples do not make sense in ecology.

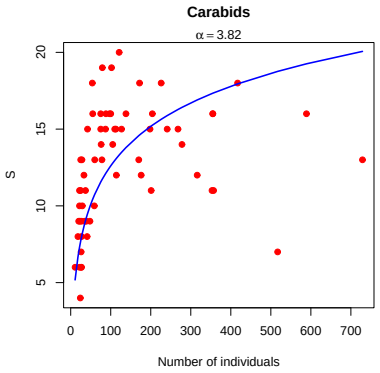
Slide 16

Species richness and sample size

Fisher log-series predicts:

$$S = \alpha \ln\left(1 + \frac{N}{\alpha}\right)$$

Species never end, but the rate of increase slows down.



Note on slide 16. Fisher's species richness model [27] can be fitted as a non-linear regression using function `nls` (library `nls`):

```
data(carabid)
S <- apply(carabid>0, 1, sum)  ## Species richness
N <- apply(carabid, 1, sum)    ## Number of carabids
fisher.fun <- function(alpha, N) {alpha * log(1 + N/alpha) }
S.fit <- nls(S ~ fisher.fun(alpha, N), start=list(alpha=8))
```

Hubbell [41] gives an almost identical approximate equation for species richness:

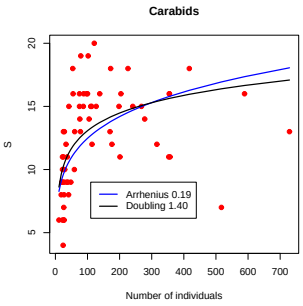
$$S \approx 1 + \theta \ln\left(1 + \frac{N-1}{\theta}\right)$$

which can be fitted just as easily as the Fisher model, and which gives almost identical results, with $\theta \approx \alpha$.

Slide 17

Species – Area models

- Island biogeography: $S = cA^z$.
- Parameter c is uninteresting, but z should describe island isolation.
- Regarded as universally good: Often the only model studied, so no alternatives inspected.
- Assuming that doubling area A brings along a constant number of new species fits often better.



Note on slide 17. The island biogeographical model derives a part of its popularity from easy fitting: It can be expressed as a log-linear model:

$$\log(S) = \log(c) + z \cdot \log(A)$$

which can be fitted in R:

```
spar.fit <- lm(log(S) ~ log(A))
b <- coef(spar.fit)
z <- b[2]
c <- exp(b[1])
```

Parameter z is the log-linear slope. It should be almost a universal constant in the range $0.15 \ldots 0.40$, and lower on islands than on the mainland. Slope z is independent of the unit of area, thanks to logarithmic transformation. Parameter c is often called an 'intercept', but this is misleading: It is the the intercept only in log-transformed equation, but not in the original exponential form — c gives the expected species richness at unit area.

An alternative model predicts a constant increase in species richness with doubling area. It can be fitted as:

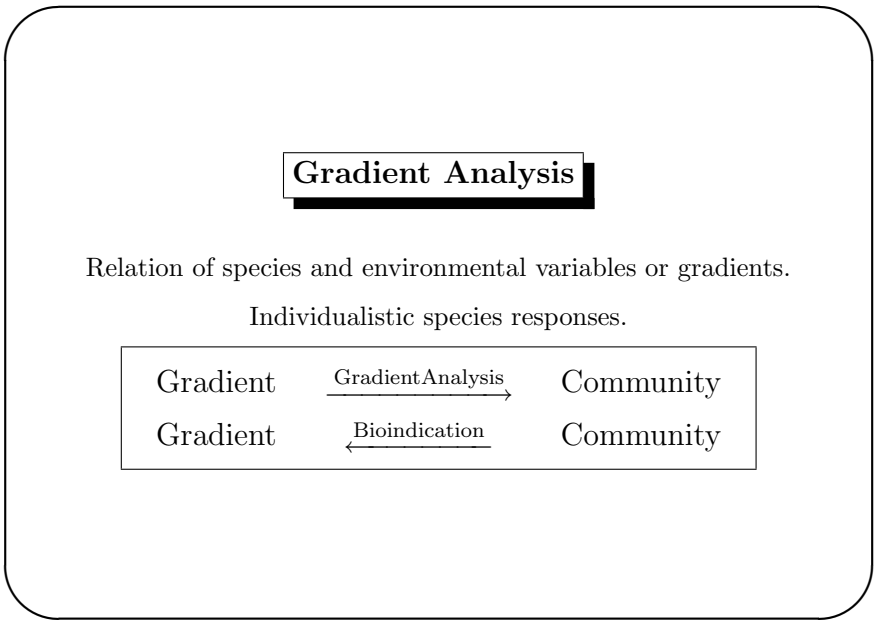
```
spar.fit2 <- lm(S ~ log2(A))
```

With base 2 logarithms, the coefficient gives directly the rate of increase per doubling A. Other bases can be used, and give the same fit, and the coefficient can be easily changed into base 2 (see note on slide 2).

2 Gradient analysis

Gradient analysis models the species abundances as a function of environmental variables or gradients. The term is often used for constrained ordination (which we discuss from slide 128 onward), but we restrict it to direct modelling of species responses.

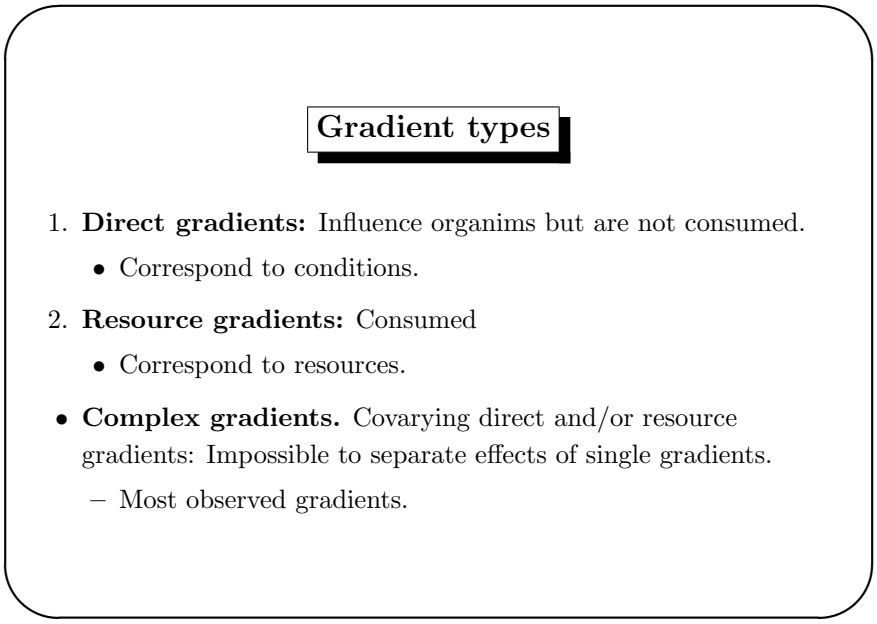
Slide 18



2.1 Basic concepts

Species responses are non-linear, but may vary in shape and among gradient types. Mike Austin [2, 3, 4, 5, 6, 7, 11] has developed the theory of response shapes.

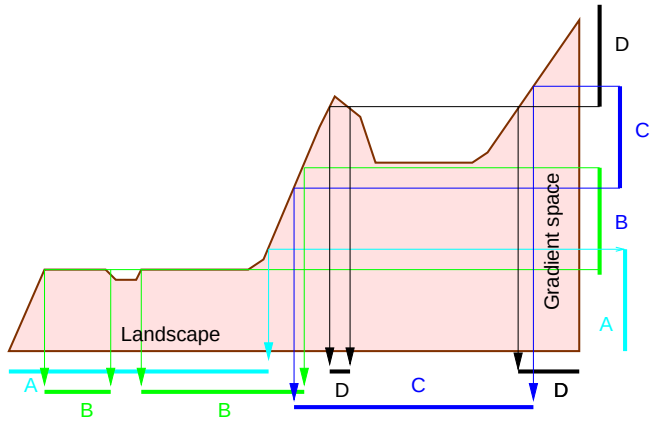
Slide 19



Note on slide 19. Based mainly on Mike Austin [7, 11].

Slide 20

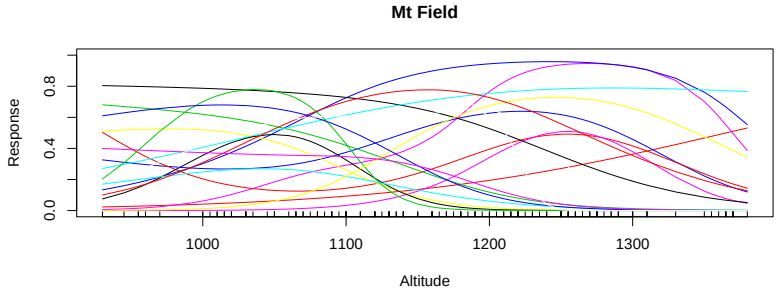
Gradients and landscape



Slide 21

Species responses

Species have non-linear responses along gradients.



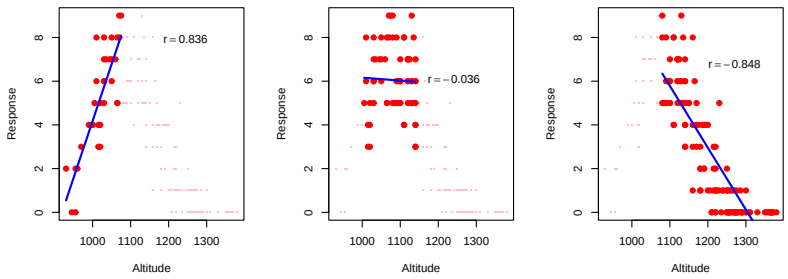
Note on slide 20. Redrawn from Austin and Smith [11].

Note on slide 21. Response models fitted as Generalized Additive Models (slide 44), degrees of freedom selected by Generalized Cross Validation.

Slide 22

Linear models are inadequate

The slope, sign and significance depend on the studied range on the gradient



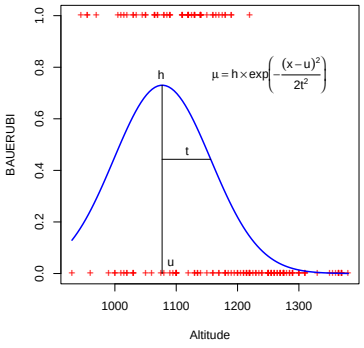
Slide 23

Gaussian response

$$\mu = h \exp\left(-\frac{(x-u)^2}{2t^2}\right)$$

Three interpretable parameters:

1. Location of optimum u on gradient x
2. Expected height h at the optimum
3. Width t of the response



Note on slide 24. Species packing is further discussed in slides 110 and 100.

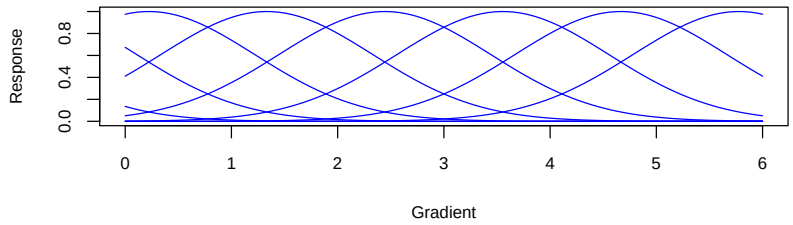
Note on slide 25. Mike Austin [3, 4, 5] has studied the confusing history of Gaussian model becoming the canonical view of our times.

Slide 24

Dream of species packing

Species have Gaussian responses and divide the gradient optimally:

- Equal heights h.
- Equal widths t.
- Evenly distributed optima u.



Slide 25

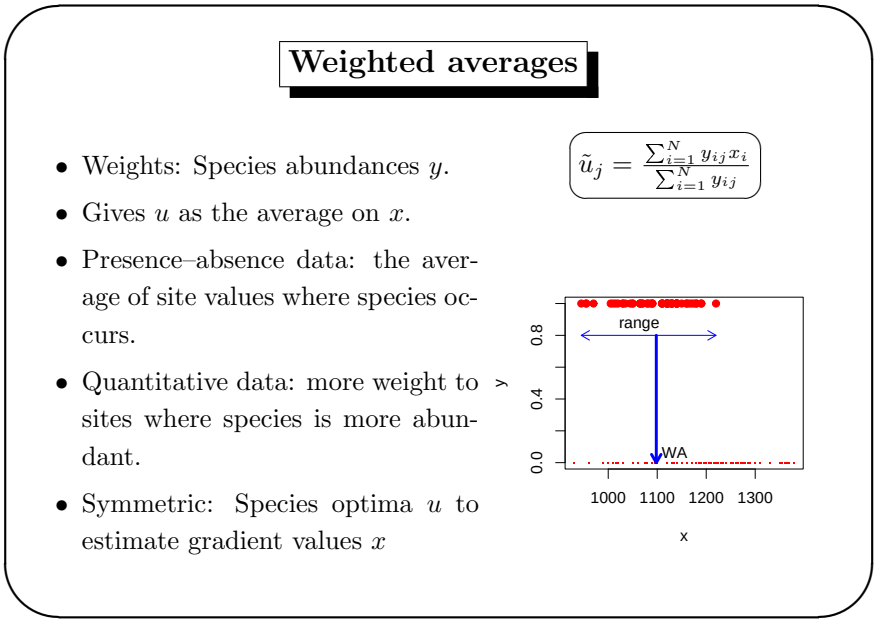
Evidence for Gaussian response

- Whittaker described many response types: multimodal, skewed, flat, plateaux and symmetric.
- Only a small part of responses were regarded as symmetric, still became the standard.
- First canonized in coenocline simulations.
- Species packing is the theoretical basis of (canonical) correspondence analysis.

2.2 Weighted averages

Weighted averages are both simpler and more adequate than traditional linear regression (slide 22). Moreover, they form a part of the theoretical justification of correspondence analysis.

Slide 26



Note on slide 26. R has a standard function `weighted.mean` which can be used directly for weighted averages. The following calculates the weighted average of Humus depth for *Cladina stellaris* in data sets `varespec` and `varechem` (which must be `attach`ed first) in library `vegan`:

```
weighted.mean(Humdepth, Cla.ste)
```

Library `vegan` has a function `wascores` which can calculate weighted averages for all environmental variables and all sites in one sweep:

```
wascores(varechem, varespec)
```

The bioindication problem of estimating gradient values x from species optima u is further discussed in slide 60.

2.3 Response models

In this section we describe briefly some of the most popular species response models and how they can be fitted to the data. Gaussian response model is the basis of the dominant school of gradient analysis [44, 87, 94]. Beta response model is used to challenge this theory [4, 10, 60]. HOF models [42] may be the best alternative to analyse response shapes [79], while GAM models [32] are the most versatile and popular alternative of gradient analysis today.

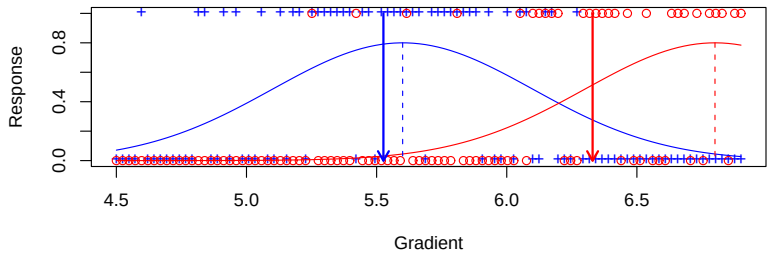
Note on slide 28. Gaussian response function and Beta response function are borrowed from statistical density functions. This has caused some confusion in the literature, since people have not noticed essential differences between a density and a response. Most importantly, density functions have a normalizing constant which guarantees that they cover a unit area (their integral is 1), whereas response functions estimate the height of response and do not have the unit area constraint. Gaussian response is sometimes even called 'Normal'

Slide 27

Bias and truncation

Weighted averages are good estimates of Gaussian optima, *unless* the response is truncated.

Bias towards the gradient centre: shrinking.



Slide 28

Popular response models

- **Gaussian response model:** The most popular model that gives symmetric responses, and is the basis of much of theory of ordination and gradient analysis.
- **Beta response:** Able to produce responses of varying skewness and kurtosis, and challenges the Gaussian dominance.
- **HOF response:** A family of hierarchic models which can be produce skewed, symmetric or different monotone responses, and can be used to analyse the response shape.
- **GAM models:** Can find any smooth shape and fit any kind of smooth response.

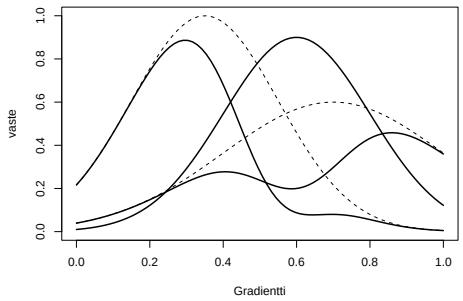
response, and its width parameter t (slide 23) is called 'sd' ("standard deviation"). A very common claim is that species appears at -2sd from its optimum and again disappears at $+2$sd from optimum so that there are no species in common at 4sd distance along a gradient [39]. It is true that 95 % of the *surface area* in Normal density function is at $\pm 2\sigma$ from the mean μ, but the expected *height* at that point is $0.135h$ which often is far higher abundance than disappearance.

In mathematical statistics [55], Beta function is denoted as B, that is uppercase Greek Beta, although ecologists often insist on lowercase β [10]. In these lecture notes I spell out the function name to avoid typographical or orthographical problems.

Slide 29

Shape matters

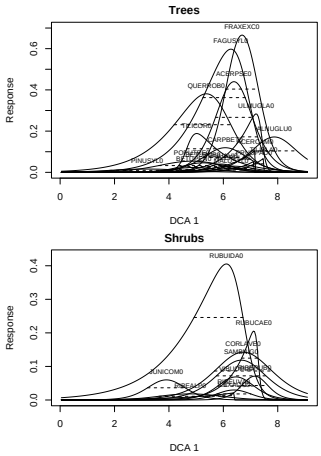
Fundamental response can be symmetric, but **realized** response skewed or multimodal due to species interactions.



Slide 30

Real World (almost)

- In Danish beech forests, dominant species skew other species away
- Austin predicted skewed responses at gradient ends
- A decent gradient would be nice instead of a DCA axis...



Note on slide 30. The graph is from [52]. It is "almost" a real world case: It uses an ordination axis instead of a real gradient.

Slide 31

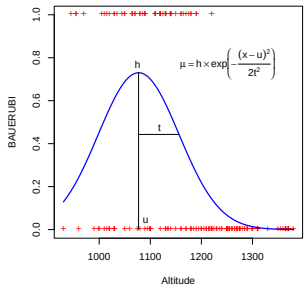
Gaussian response: a case of GLM

Can be reparametrized as a generalized linear model:

$$\begin{aligned} \mu &= h \exp \frac{(x-u)^2}{-2t^2} \\ \log(\mu) &= b_0 + b_1 x + b_2 x^2 \end{aligned}$$

- Gradient as a 2nd degree polynomial.
- Logarithmic link function.

$$\begin{aligned} u &= -\frac{b_1}{2b_2} \\ t &= \sqrt{-\frac{1}{2b_2}} \\ h &= \exp\left(b_0 - \frac{b_1^2}{4b_2}\right) \end{aligned}$$



Note on slide 31. Generalized linear models are fitted with function `glm` in R. In the following, I use generic names `y` for species abundance vector, and `x` for the environmental variable. We need logarithmic link function, which is the default with Poisson error that may be adequate for ecological abundance data in many cases:

```
mod <- glm(y ~ x + I(x^2), family=poisson)
b <- coef(mod)
u <- -b[2]/2/b[3]
t <- sqrt(-1/2/b[3])
h <- exp(b[1] - b[2]^2/4/b[3])
```

In R we must Isolate the second degree term with `I(x^2)`, because R would otherwise compute the sum $x + x^2$ and use that as a single explanatory variable instead of separate first and second degree terms.

R has a function `poly` which will directly produce polynomials of given degree, and so we can in some cases use the formula `y ~ poly(x,2)`. Function `poly` produces orthonormal polynomials, and we cannot use it (easily) if we want to get back the Gaussian parameters. However, if we are interested only in the fitted values, it is better to use `poly` than the explicit polynomials.

The use of polynomial with logarithmic link function is not an approximation, but it will give exact (within numerical accuracy) estimates of the Gaussian parameters and correct confidence intervals for the fitted values. However, the confidence intervals of the Gaussian parameters need more involved calculations [77, 93].

Note on slide 32. Full introduction into Generalized Linear Models is beyond these Lecture Notes: They deserve their own lectures. I assume that the basic concepts are familiar — or shall become familiar.

The fundamental reference to GLM is McCullagh & Nelder [57] which is very readable even for a layman. For an ecologist, Crawley [18] is a very good introduction, although it is targeted to an obsolescent program.[1] Venables & Ripley [97] are true to their style and very laconic. In addition, there is an abundant on-line documentaion on the web and with the standard installation for R.

[1] He seems to have updated this for S-plus, but I have not yet studied the book.

Slide 32

Generalized linear models: a refresher

1. **Linear predictor** η: a linear function of explanatory variables, which can be continuous or classes, and can be transformed variables, or powers or polynomials

$$\eta = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p$$

2. **Link function** $g(\cdot)$ that transforms the fitted values μ to the linear predictor η

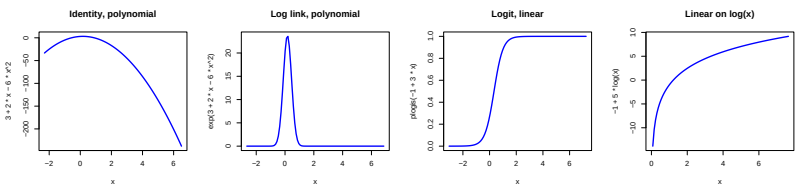
$$g(\mu) = \eta$$

3. **Error distribution** from the exponential family to describe the distribution of residuals about fitted values.

Slide 33

Special cases of GLM

Model	Link	Error	Variance
Linear model	Identity $\mu = \eta$	Normal	Constant
Log-linear	Logarithmic	Poisson	μ
Logistic	Logistic	Binomial	$\mu(1-\pi)$



Note on slide 33. The concept of linearity is different from the common sense, since many a linear model may be curved. The key phrase is 'linear in parameters', meaning that the linear predictor is a *sum* of *products* or $\sum(b \times x)$.

Slide 34

Ecologically meaningful error distributions

Normal error rarely adequate in ecology, but GLM offer ecologically meaningful alternatives.

- **Poisson.** Counts: integers, non-negative, variance increases with mean.
- **Binomial.** Observed proportions from a total: integers, non-negative, have a maximum value, variance largest at $\pi = 0.5$
- **Gamma.** Concentrations: non-negative real values, standard deviation increases with mean, many near-zero values and some high peaks.

Slide 35

Goodness of fit and inference

- Deviance: Measure of goodness of fit
  - Derived from the error function: Residual sum of squares in Normal error
  - Distributed approximately like χ^2
- Residual degrees of freedom: Each fitted parameter consumes one degree of freedom and (probably) reduces the deviance.
- Inference: Compare change in deviance against change in degrees of freedom
- Overdispersion: Deviance larger than expected under strict likelihood model
- Use F–statistic in place of χ^2.

Note on slide 35. Overdispersion is the rule in ecology. Naïve followers of textbooks customarily perform χ^2 tests and get crossly wrong results. Please note that the F statistic indeed is *defined* as a product of scaled χ^2 variates, and that even χ^2 variates are derived from Normal distribution instead of being somehow "non-parametric" [55]. If x is distributed Normally with mean μ and variance σ^2 or $x \sim N(\mu, \sigma^2)$, we can transform it to a standard Normal variate $z \sim N(0, 1)$ using $z = (x - \mu)/\sigma$. The χ^2 distribution is *defined* as the sum of squared standard Normal variates or $\sum_{i=1}^{p} z_i^2 \sim \chi^2(p)$ — nothing "non-parametric" so

far. Fisher's statistic is the ratio of two scaled χ^2 variates:

$$F_{p,q} = \frac{\chi^2(p)/p}{\chi^2(q)/q}$$

This derivation explains why we use χ^2 distribution to evaluate a likelihood based test statistic, deviance, and why we can rescue our case with switching to F in case of overdispersion [1, 57, 97].

R has very practical 'quasifamilies' `quasipoisson` and `quasibinomial` for overdispersed data. These automatically adjust the standard errors of coefficients, and do the correct F test in `anova` (the user must request this test statistic). In addition, there is the generic `quasi` family which requests the link and variance functions [97]. The following modifies the simplistic approach of slide 31, this time with a real data, where we study whether we really need a Gaussian model for *Cladina stellaris*:

```
Claste.glm <- glm(Cla.ste ~ Humdepth + I(Humdepth^2), family=quasipoisson)
anova(Claste.glm, test="F")
Analysis of Deviance Table

Model: quasipoisson, link: log

Response: Cla.ste

Terms added sequentially (first to last)

             Df Deviance Resid. Df Resid. Dev      F  Pr(>F)
NULL                            23     905.92
Humdepth      1   144.59        22     761.33 3.6998 0.06808 .
I(Humdepth^2) 1     0.24        21     761.09 0.0061 0.93855
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

No, we don't need the second degree term, but we could do with the first degree alone (and even that is not too important).

The Poisson error in the strict likelihood accounts only for the sampling error in y in repeated studies in the same site. However, there are other sources of error variation:

- We do not know all important environmental variables that influence the abundances, but their effect is visible only as inflated variation.
- Biological interactions, clonal dispersal, source–sink dynamics increase the error.

This means that the expected value μ is itself a random variate with its own error distribution. This calls for modes with two error components. Some of these are well-known and much studied:

Error in y	+ *Error in μ*	= *Resulting distribution*
Poisson	+ Gamma	= Negative Binomial
Binomial	+ Beta	= Beta-Binomial

Library `MASS` [97] has function `glm.nb` for fitting Negative Binomial GLMs — for other cases we have to wait or write the function ourselves. The following references discuss compound error distributions: [1, 17, 48, 64, 76].

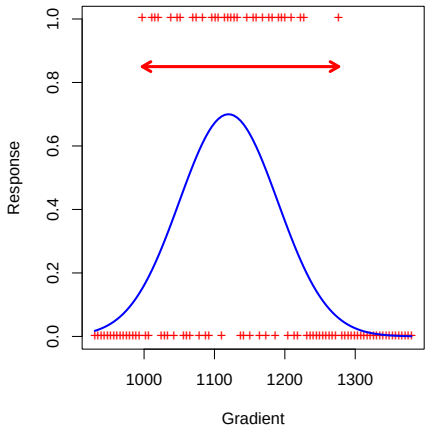
Note on slide 36. Gaussian response is occasionally criticized for predicting every species to occur everywhere at low abundances, whereas real species have absolute limits of range [9, 10]. This argument confuses expected responses (which may not have limits) and observed responses (which may still have limits), like the slide shows.

Some people confuse Gaussian response function with Normal probability density function. Because 95 % of the *area* covered by a Normal PDF is at $-2\sigma \ldots + 2\sigma$, they say that species appear at $u - 2t$ and disappear at $u + 2t$. The response function does not have a unit area constraint, and there we should look at the *height* at a point instead of the area covered. The issue is further discussed in slide 56 and by [33].

Slide 36

Gaussian model and response range

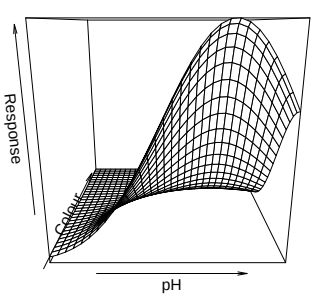
- Gaussian response is never exactly zero: Asymptotic model
- Observed abundances have a discrete component
- The observed range depends on parameters t and h

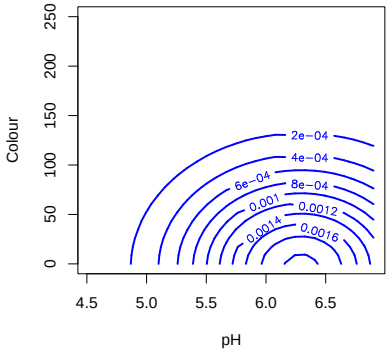


Slide 37

Several gradients

- Gaussian response can be fitted to several gradients: Bell

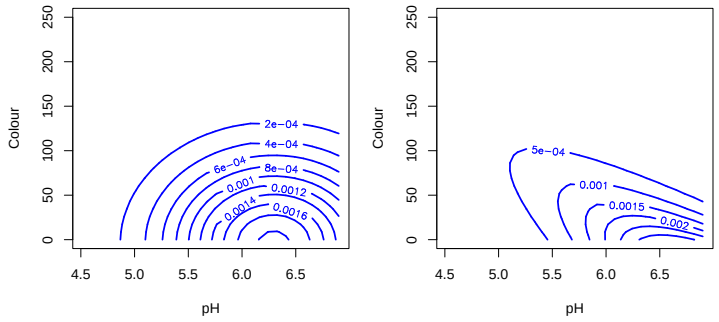




Slide 38

Interactions in Gaussian responses

- No interactions: Responses parallel to the gradients
- Interactions: The optimum on one gradient depends on the other



Note on slide 38. Two-dimensional Gaussian responses are very easy to fit:

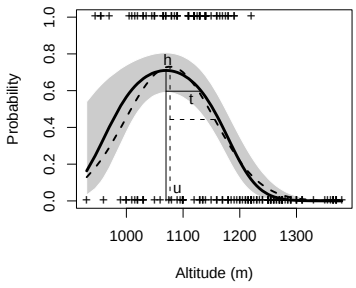
```
> glm(y ~ x + I(x^2) + z + I(z^2), family=poisson, ...)
> glm(y ~ x + I(x^2) + z + I(z^2) + x*z, family=poisson, ...)
```

The first model defines a model without interaction (slide 37), and the second model defines the full model of this slide. The Gaussian parameters are somewhat more tricky to find, in particular with interaction terms; see [44, 77] for techniques in different cases.

Slide 39

Logistic Gaussian response

- Polynomial often used with other link functions than log.
- Binomial error: logistic link.
- The Gaussian parameters correct only with log link: Width t has different interpretation.



Note on slide 39. The polynomial predictor in GLM is so ubiquitously used that the *polynomial* has become synonymous with the Gaussian response. However, it is strictly equal with Gaussian only with the logarithmic link [79]. The following model is not strictly Gaussian:

```
glm(y ~ x + I(x^2), family=binomial)
```

because the default link function with Binomial error is logistic:

$$\mu = \frac{\exp \eta}{1 + \exp \eta}$$

instead of the simple $\mu = \exp \eta$ with logarithmic link. The general bell shape is similar, but the response becomes increasingly flat as height h approaches the binomial denominator. The width parameter t defines a smaller decrease in abundance as h increases toward the upper limit. Although the interpretation of t changes, its calculation remains unchanged. Optimum u has the same meaning and same calculation as with the strict Gaussian model, although it may move somewhat because the fitted model changes. The calculation of height must be adopted to the changed link function. The following will always use the correct inverse link function in R:

```
h <- Claste.glm$family$linkinv(b[1] - b[2]^2/4/b[3])
```

Compare this to the slide 31. The form above is too cumbersome for normal interactive use, since we usually know a shorter name for the inverse link function, but it is practical for a general function. The first part of the function (`Claste.glm`) must be the name of a `glm` object that we have fitted previously.

In R it is permissible to use logarithmic link function with Binomial error, and in principle, we can write:

```
glm(y ~ x + I(x^2), family=binomial(link="log"))
```

However, this usually fails to fit. We need extra control of the fitting process, or perhaps easier, we must use non-linear maximum likelihood regression. I won't explain this here, but the procedure becomes familiar later (slide 40).

The example with Binomial error above is valid only for binary observations: for general Binomial case we need to supply the Binomial denominator. In R the recommended way is to use a two-column matrix with 'success' and 'failure' as the dependent variable. If y is the number of occurrences (for instance, hits in a point frequency frame) for a species, and *tot* is the total number of trials (total number of points in a point frequency frame), the model:

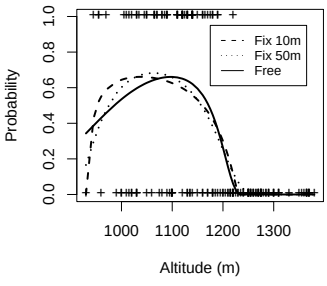
```
glm(cbind(y,tot-y) ~ x + I(x^2), family=binomial)
```

Slide 40

Beta response

$$\mu = k(x - p_1)^{\alpha}(p_2 - x)^{\gamma}$$

- Responses with varying skewness and kurtosis.
- Simulated coenoclines to test robustness of ordination.
- Commonly fitted fixing endpoints p_1 and p_2 and using GLM: Not flexible any longer, but greatly influenced by endpoints.
- **Must** be fitted with non-linear regression.



Note on slide 40. The customary way of fitting beta response function is based on logarithmic transformation [8, 10, 47]:

$$\begin{aligned} \mu &= k(x-p_1)^{\alpha}(p_2-x)^{\gamma} \\ \log(\mu) &= \log(k)+\alpha\log(x-p_1)+\gamma(p_2-x) \end{aligned}$$

so that the function becomes a special case of generalized linear models:

```
x1 <- log(x - p1)
x2 <- log(p2 - x)
beta.mod <- glm(y ~ x1 + x2, family=poisson)
```

This defines a GLM *conditional* to parameters p_1 and p_2, or 'endpoints', which were fixed before fitting. However, after fixing, the beta response function is no longer versatile, and it is impsosible to estimate (1) the location of the optimum, (2) skewness and (3) kurtosis, or three characteristics, using only two parameters α and γ [74]. Austin [9] would like to see the fixing of endpoints as an ecological problem, but that is not technically possible with beta function which is strongly dependent on the arbitrary fixed values, and just doesn't work like desired.

The only reasonable way of fitting the beta function is to estimate simultaneously all five parameters [74, 79]. This can be done with non-linear regression. In R we can fit non-linear least squares models with function `nls` (library `nls`) and maximum likelihood models with function `nlm`. The latter is slightly more awkward to use, but vastly more useful, and therefore I explain only its use.

I explain here fitting a binary model (Binomial with denominator = 1), but the model is easily adopted to other cases. Analogously to Gaussian polynomial (slide 39), I base the model on the log transformed 'linear predictor', but use logistic link so that the fitted values are guaranteed to be in the range $0 \ldots 1$. With beta response function this is better justified than in the Gaussian model, since the parameters of the beta function are not directly interpretable or interesting (see slide 41). We want to analyse both the general case and the case where the response is strictly symmetric or $\alpha = \gamma$. We must give the parameters $(p_1, p_2, k, \alpha, \gamma)$ in one parameter vector in R. The following function will satisfy these conditions:

```
beta.fun <- function(p, x)
{
  zero <- .Machine$double.eps
  one <- 1 - zero
  if (length(p)==4) p <- c(p,p[4])
  eta <- p[1] + p[4]*log(x-p[1]) + p[5]*log(p[2]-x)
  fv <- plogis(eta)
  fv[x <= p[1]] <- ifelse(p[4] < 0, one, zero)
  fv[x >= p[2]] <- ifelse(p[5] < 0, one, zero)
  fv
}
```

We need to supply good guesses of starting values in non-linear estimation. A good choice is to use GLM with fixed endpoints:

```
beta.start <- function(x, y, extend=50, symmetric=FALSE)
{
  p1 <- min(x[y>0]) - extend
  p2 <- max(x[y>0]) + extend
  inc <- x > p1 & x < p2
  if (symmetric)
    p <- coef(glm(y ~ I(log(x-p1)+log(p2-x)), family=binomial, subset=inc))
  else
    p <- coef(glm(y ~ log(x-p1) + log(b-x), family=binomial, subset=inc))
  p <- c(p1,p2,p)
  p
}
```

This puts the 'endpoints' p_1 and p_2 at `extend` units from the extreme occurrences. Would we believe that it is adequate to fit beta response as a GLM, this function would be sufficient for us.

Now we have the function to produde the fitted values and starting values. We need only the likelihood function:

```
beta <- function(p, x, y)
{
  fv <- beta.fun(p, x)
  -sum(dbinom(y, 1, fv, log=TRUE))
}
```

and we can fit the maximum likelihood regression with:

```
sol <- nlm(beta, p=beta.start(Altitude, BAUERUBI), x=Altitude, y=BAUERUBI)
```

The example uses species `BAUERUBI` and `Altitude` gradient. Future versions of `gravy` should have a more user-friendly way of fitting Beta functions.

Slide 41

Parameters of Beta response

- No clearly interpreted parameters.
- α and γ define: $\mu = k(x - p_1)^\alpha (p_2 - x)^\gamma$
  1. The location of the mode.
  2. The skewness of the response.
  3. The kurtosis of the response.
- Response is zero at p_1 and p_2: absolute endpoints of the range.
- k is a scaling parameter: height depends on other parameters as well.

Note on slide 42. The 'HOF' models are named after their inventors Huisman, Olff and Fresco [42]. These authors provide a proprietary program for fitting the models with least squares. The HOF models can be fitted using R [79], but this is fairly tedious. I have a special document about fitting HOF models on my web pages. Moreover, experimental package `gravy` has a canned function for fitting HOF models, although it is still very limited in choices of error distribution (at version `0.0-11`).

Library `gravy` can fit HOF models either for a single species or for all species in a `data.frame` in a single pass:

```
> HOF(BAUERUBI, Altitude, M = 1)

Call:
HOF.default(spec = BAUERUBI, grad = Altitude, M = 1)

BAUERUBI
            a         b         c         d code
V   -19.54340  34.55569   0.29265   2.42751
IV   -5.63548  11.52299   1.94475
```

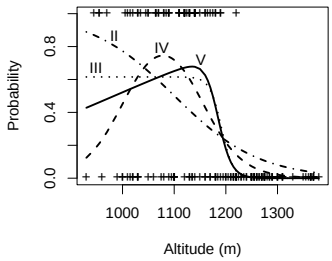
Slide 42

HOF models

$$\mu = \frac{M}{[1+\exp(a+bx)]\cdot[1+\exp(c-dx)]}$$

Huisman–Olff–Fresco: A set of five hierarchic models with different shapes.

	Model	Parameters			
V	Skewed	$\underline{a}$	$\underline{b}$	$\underline{c}$	$\underline{d}$
IV	Symmetric	$\underline{a}$	$\underline{b}$	$\underline{c}$	b
III	Plateau	$\underline{a}$	$\underline{b}$	$\underline{c}$	∞
II	Monotone	$\underline{a}$	$\underline{b}$	0	0
I	Flat	$\underline{a}$	0	0	0




```
III -23.22468  39.98669  -0.46608
II   -2.07938   5.49397
I     0.63090
                     V       IV      III       II        I
deviance:    137.5181 150.0717 139.1160 170.8117 215.6847
residual df: 163.0000 164.0000 164.0000 165.0000 166.0000
AIC:         143.5181 154.0717 143.1160 172.8117 215.6847

> HOF(mtf01, Altitude, 1)

              Df    Dev      a     b     c   d
EPACSERP V   163 210.25 -19.71 22.75  1.55 3.1
CYATPETI IV  164 122.60 -10.84 13.74  8.80
NOTHCUNN IV  164  89.74  -6.10 17.72  3.79
POA.GUNN IV  164 206.11  -5.66  7.37  2.97
BAUERUBI III 164 139.12 -23.22 39.99 -0.47
```

Note on slide 44. GAMs are immensely useful for ecologists even outside gradient modelling. They can used both for the seeking of the real forms without forcing the model into a linear framework, or they can be used to verify the adequacy of a parametric model. The basic reference is [32], but most introductions to R or S-plus cover GAMs as well [97]. The GAMs are very popular in gradient modelling [13, 23, 34, 53, 79, 109, e.g.].

The GAMs are generalized from GLM (slide 32), but they replace the linear predictor with a non-linear smoother. Almost everything else remains similar as in GLM: Error distributions of the exponential family, link function, model testing etc.

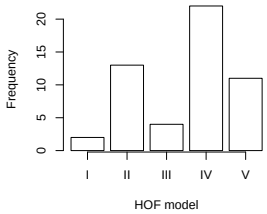
GAMs are best fitted with library `mgcv` in R [107, 108]. The basic usage is very simple:

```
> gam(BAUERUBI ~ s(Altitude), family=binomial)

Family: binomial
Link function: logit

Formula:
BAUERUBI ~ s(Altitude)

Estimated degrees of freedom:
```

Slide 43

HOF: Inference on response shape

- Alternative models differ only in response shape.
- Selection of parsimonous model with statistical criteria.
- 'Shape' is a parametric concept, and parametric HOF models may be the best way of analysing differences in response shapes.



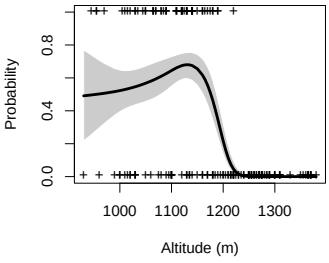
Most parsimonous HOF models on Altitude gradient in Mt. Field, Tasmania.

Slide 44

Generalized Additive Models (GAM)

- Generalized from GLM: linear predictor replaced with smooth predictor.
- Smoothing by regression splines or other smoothers.
- Degree of smoothing controlled by degrees of freedom: analogous to number of parameters in GLM.
- Everything else like in GLM.
- Enormous use in ecology — also outside gradient modelling.

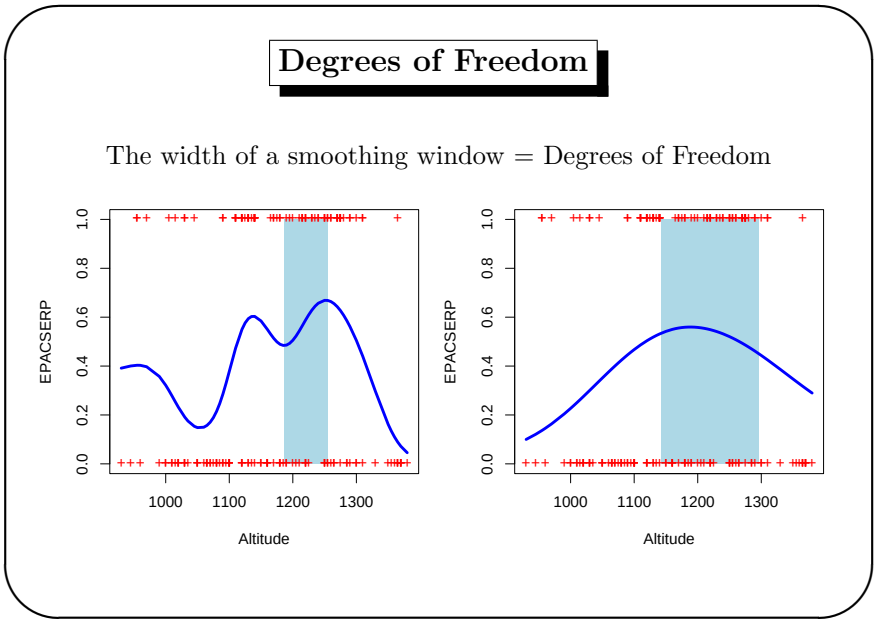
$$g(\mu) = \text{smooth}(x)$$



```
 3.188498   total =  4.188498

UBRE score:  -0.2884196
```

The best introduction to the usage in R is [108].

Slide 45



Note on slide 45. Degrees of freedom used by the regression define how sensitive the fitted response is to local features in the data. In general, we may think that the more we use degrees of freedom, the narrower is the neihbourhood influencing the fit. The R function `gam` in `mgcv` uses regression splines, and there the smoothness is defined by the number of knots [32, 107, 108] instead of a window, but the implication is similar.

Function `gam` of the library `mgcv` selects the degrees of freedom using generalized cross validation [107]. Consequently, they are usually real numbers (see the example with slide 44). However, the user can define either an upper limit for the search or fix the degrees of freedom [108]. The latter practice is similar as in the function `gam` of S-plus [32]. When the degrees of freedom are fixed, alternative models are compared using ANOVA-like procedures. With cross-validatory selection of the degrees of freedom, this is not necessary. Simon Wood [108] gives more detailed guidelines for model building in R.

Note on slide 46. The standard plot function uses the `link` scale, which actually was used in fitting. Functions `fitted`, and optionally `predict(..., type="response")` will use the original scale of responses. The upper and lower confidence limits must be first calculated in the `link` scale and then transformed to the `response` scale with the inverse link function (available in `family$linkinv`).

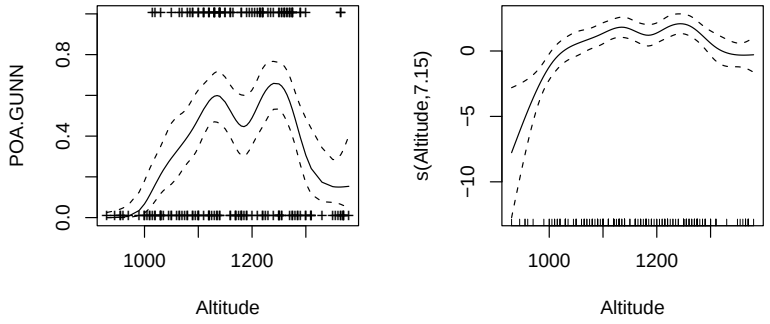
Note on slide 47. The multidimensional GAM is easily fitted:

```
> gam(cbind(Neidaffi, tot-Neidaffi) ~ s(pH) +  s(Colour), family=binomial)

Family: binomial
Link function: logit

Formula:
cbind(Neidaffi, tot - Neidaffi) ~ s(pH) + s(Colour)

Estimated degrees of freedom:
```

Slide 46

Linear scale and response scale

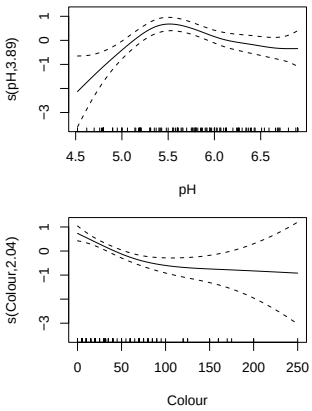
GAM is smooth in the link scale, but the user prefers the response



Slide 47

Multiple gradients

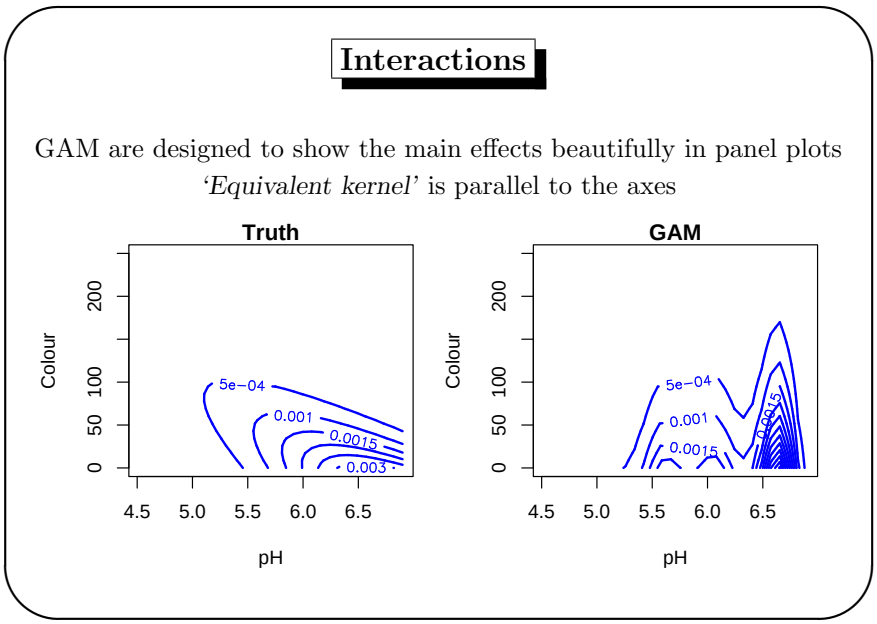
- Each gradient is fitted separately
- Interpretation easy: Only the individual main effects shown and analysed
- Possible to select good parametric shapes
- Thin-plate splines: Same smoothness in all directions and no attempt of making responses parallel to axes



```
 3.887356 2.042436   total =  6.929793

UBRE score:  1.264500
```

Here we used the same way of handling binomial responses as in the slide 39. The graph is produced with the standard `plot` command which uses the scaling of the link function and adds the approximate confidence intervals to the graph.

Slide 48



Note on slide 48. The GAMs are *designed* to show the main effects only, because this is what people want. For surfaces on multiple gradients this means that you take the marginal effects of each environmental variable and add their effects (GAM are *additive* models). This may be the most dangerous feature of GAMs in gradient analysis. Hastie and Tibshirana [32] describe both this problem with the *equivalent kernel* and discuss analysing the interactions. No very good alternatives are found for the latter.

The simulation was made on a diatom data set using the real sampling pattern. The responses are modelled after *Asterionella formosa*.

2.4 Beta diversity and scaling of gradients

The rate of change of community along an ecological gradient is known as beta diversity. The total change over a gradient interval gives an ecologically meaningful estimate of the importance of the gradient, or the gradient length. Further, we may wish to scale the gradient so that the beta diversity is constant in all gradient locations.

Note on slide 49. Whittaker presented his concepts in various and sometimes contrasting ways [85, 99, 100, 101]. Alpha (α) diversity is the ordinary diversity of community, measured either as a diversity index (slide 2) or species richness (slide 14). Beta (β) diversity is defined in various ways which is the subject of the remaining chapter. The other componetns are largely forgotten.

Note on slide 51. "Whittaker's index" is a common name in English language literature, but the index was used much earlier in continental phytosociology, where it was called Klement's index [50]. Barkman [12] even suggested a cure to dependence on sample size: count only species exceeding a certain frequency (e.g., 10 %) into S_{TOT}.

Slide 49

Diversity and spatial scale

- Whittaker suggested several concepts of diversity
  - α: Diversity on a sample plot, or 'point' diversity.
  - β: Diversity along ecological gradients.
  - γ: Diversity among parallel gradients or classes of environmental variables.
  - δ: The total diversity of a landscape: sum of all previous.

Slide 50

Many faces of beta diversity

What are we talking about when we are talking about beta diversity?

1. General heterogeneity of a community.
2. Decay of similarity with gradient separation.
3. Widths of species responses along gradients.
4. Rate of change in community composition along gradients.

Slide 51

General heterogeneity

- "Whittaker's index": Proportion of average species richness on a single plot $\bar{S}$ and thet total species richness in all plots S_{TOT}.
- Total richness increases with increasing sample size.
- Average richness stabilizes with increasing sampling effort.
- $\bar{S}/S_{\mathrm{TOT}}$ decreases with sample size.
- No reference to gradients: even with a single location, replicate sampling decreases the index.
- Pattern diversity: Within site diversity.

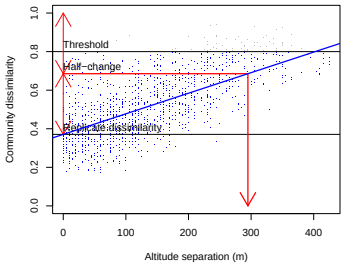
Pattern diversity is a rarely used term which refers to the heterogeneity within one stand [110]. The general heterogeneity index may indeed be rather related this concept. Hubbell's [41] concept of beta diversity is most closely related to pattern diversity, although he seems to prefer the approach of slide 52.

Slide 52

Similarity decay with gradient separation

Plot community (dis)similarity against gradient separation and fit a linear regression.

- Intercept: (Dis)similarity at zero-distance – 'noise', replicate (dis)similarity, general heterogeneity or pattern diversity.
- Slope: Beta diversity.
- Half-change: Gradient ditance where expected similarity is half of the replicate similarity (intercept).



Note on slide 52. The figure was drawn using function `postMDS` in library `vegan`. This function is intended for scaling of ordination axes in half-scale units [62], but it works for ordinary gradients as well. Community (dis)similarity is linearly related to gradient separations only below some dissimilarity threshold, and the regression should be fitted only within this linear region.

The similarity decay was a popular measure of beta diversity in olden days [101]. It resurrected with biodiversity studies [65], and Hubbell [41] uses this index in describing the beta diversity of a metacommunity landscape. Hubbell's beta diversity refers to dispersal

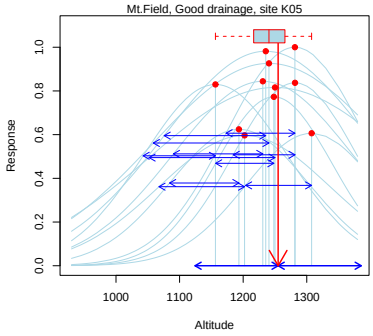
friction in biogeographical scale within homogeneous environment.

Slide 53

Hill indices of beta diversity

1. Average width of species responses.
2. Variance of optima of species occurring in one site.

- Used with scaling of ordination axes.
- The first index discussed and described, but the second applied.
- Equal only to degenerated species packing gradients.



Gaussian responses fitted to species occurring in one site.

Note on slide 53. The measures were originally developed for scaling ordination axes to constant beta diversity [38, 39]. Hill and Gauch [39] discuss only the first Hill index, the average width of species responses. In the program manual [38], Hill mentions briefly that he actually used Hill index 2, the variance of species optima, instead of the better known index.

Slide 54

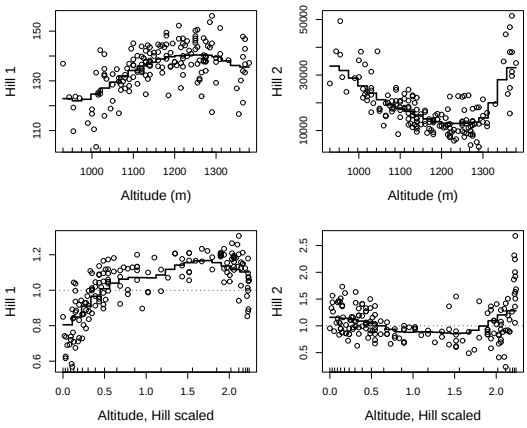
Hill scaling in practice

- Hill index spaced on species occurrences in sites: random variation.
- Smoothed by segments.
- Each segment made equally long in terms of the Hill index — almost...
- Four cycles commonly performed, but not enough to stabilize the Hill index (with half steps taken).

Note on slide 55. Hill rescaling is generally available for ordination axes only (see slide 99). Package `gravy` has function `hillscale` for scaling real gradients. The plot was produced with this function.

Slide 55

Hill rescaling of gradients



Slide 56

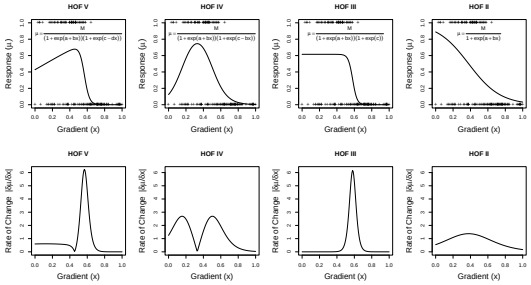
Are there species in common at '4sd' distance?

- Confounds Normal probability density and Gaussian response: Density had 95 % of its survace at $\mu \pm 2\sigma$, but the height of the response is $0.135h$
- The range of species depends on h, but in many cases a more realistic limit is $u \pm 3t$, where $\mu = 0.01h$
- If widths t vary, some species occur at longer distances.
- Look at your data before saying that there are no species in common at 4 'sd'.

Slide 57

Rate of change along gradients

Instantaneous rate of change δ at any gradient point x estimated from fitted species response functions μ:



$$\delta(x) = \sum_{j=1}^{S} \left| \frac{\partial}{\partial x} \mu_j(x) \right|$$

Note on slide 57. Oksanen and Tonteri [80] suggested using fitted responses in evaluating the rate of change. The index had a long history earlier, but it was based on actual species appearances and disappearances or species turnover [105, 106]. However, much of this turnover along ecological gradients is caused by random variation and it is better called as pseudoturnover [96]. With response function we do the smoothing before assessing the turnover, and avoid pseudoturnover. On the other hand, if our response function is wrong — biased or too rigid — we may get badly biased rate of change.

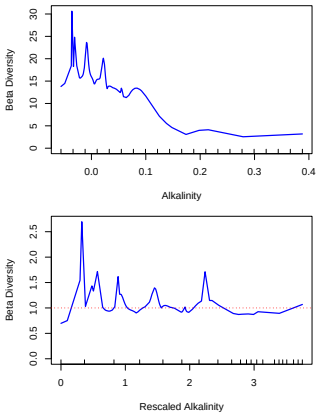
Slide 58

Rescaling to constant rate of change

- Make interval between any two gradient points a and b equal to the total accumulated change Δ_{ab} between points:

$$\Delta_{ab} = \int_a^b \delta(x) \cdot \mathrm{d}x$$

- Can be based on any response model: The example uses HOF.



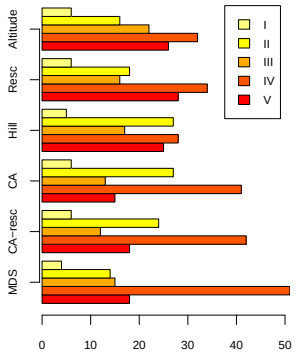
Note on slide 58. The plot was produces using package `gravy` which has function for fitting HOF models (`HOF`), evaluating the beta diversity from fitted responses at any gradient point (`betadiversity`) and for scaling the gradient to the constant rate of change (`gradscale`).[1]

[1] The plot requires version `0.1-0` of the package which was not yet publically available when writing this.

Slide 59

Alternative rescaling and response shapes

- Direct rescaling and Hill rescaling are inconsistent.
- Two Hill indices of beta diversity are inconsistent.
- None of the rescaling methods produce symmetric response shapes.
- Ordination axes tend to produce symmetric responses.



Note on slide 59. Following Økland [68], many people say that Hill rescaling produces gradients with symmetric (and equally wide) species responses, and so avoids some problems with observed gradients. Peter Minchin [63] claims that Økland's results were an artefact of using a correspondence analysis axis, and not by rescaling *per se*. It seems that Minchin is correct: none of the rescaling methods produces symmetric species responses. On the other hand, non-metric multidimensional scaling produces just as symmetric responses as correspondence analysis.

2.5 Bioindication

In bioindication we use species optima or species indicator values to obtain an estimate of unknown environmental conditions, or gradient values. The method is known with many alternative names: Calibration, reconstruction of palæoenvironments *etc.*.

Slide 60

Weighted averages in bioindication

$$\tilde{x}_i = \frac{\sum_{j=1}^{S} y_{ij} u_j}{\sum_{j=1}^{S} y_{ij}}$$

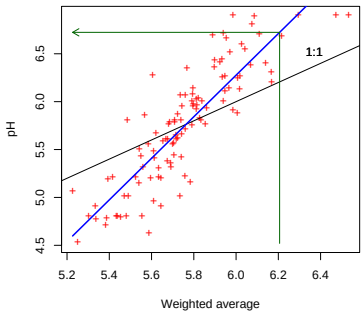
- Weighted average of indicator values of species occuring in a site.
- Can use species weighted averages $\tilde{u}_j$ or other indicator values u_j.
- Repeated cycling $x \to \tilde{u}, \tilde{u} \to \tilde{x}, \ldots, \tilde{x} \to \tilde{u}$ gives a solution of first axis in correspondence analysis.
- The range and variance of weighted averages is smaller than the range of values they are based on: "deshrinking" to restore the original variance.

Note on slide 60. Weighted averages may sound simple, but it has been found one of the most reliable method in bioindication and it is commonly used in diatom based environmental calibration [14, 75, 91, 93, 95]. It is common to use species weighted averages $\tilde{u}$ in bioindication, but other indicator values may be even more useful, since they avoid the circularity. For instance the Ellenberg indicator values [24] have been found useful [83] and consistent [92].

Slide 61

Deshrinking: stretch weighted averages

1. Inverse regression: regress gradient values on WAs.
2. Classical regression: regress WAs on gradient values.
3. Simple stretching: make variances equal.



Note on slide 61. Birks et al. [14] have studied alternative ways of deshrinking, or improving the correspondence between weighted averages $\tilde{x}$ and real gradient values x. They proposed two methods based on regression analysis. Inverse regression (Figure on slide 61) is simpler:

$$\begin{aligned} E(x) &= b_0 + b_1\tilde{x} \\ \tilde{x} &\Leftarrow E(x) \end{aligned}$$

Classical regression is more natural in its choice of dependent and independent variables, but a bit more cumbersome to use:

$$\begin{aligned} E(\tilde{x}) &= b_0 + b_1 x \\ \tilde{x} &\Leftarrow (\tilde{x} - b_0)/b_1 \end{aligned}$$

Inverse regression directly minimizes the prediction residuals, and gives an apparently lower prediction errors than classical regression.

Both are easily performed in R. We shall first get the weighted averages with function `wascores` in library `vegan` using variable pH and dataset on diatoms:

```
dia.wa <- wascores(pH, dia)       # WA scores of species
pH.wa <- wascores(dia.wa, t(dia)) # WA predictions for sites
pH.invreg <- lm(pH ~ pH.wa)       # Inverse regression
pH.wa1 <- predict(pH.invreg)
pH.clareg <- lm(pH.wa ~ pH)       # Classical regression
b <- coef(pH.clareg)
pH.wa2 <- (pH.wa-b[1])/b[2]
```

These regression methods remove both the bias and error (see slide 62), whereas equalizing variances makes only deshrinking without general bias adjustment. In R this can be done with a one-liner:

```
pH.wa3 <- sd(pH)/sd(pH.wa)*(pH.wa-mean(pH.wa)) + mean(pH.wa)
```

Slide 62

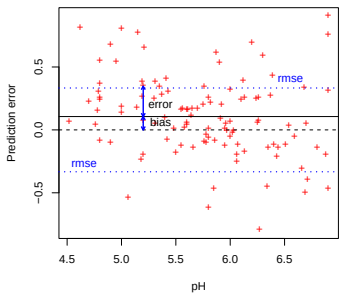
Goodness of prediction: Bias and error

Goodness: prediction error.

Correlation bad: depends on the range of observations.

- Root mean squared error $\epsilon = \sqrt{\sum_{i=1}^{N}(\tilde{x}_i - x_i)^2/N}$.
- Bias b: systematic difference.
- Error ε: random error about bias
- $\epsilon^2 = b^2 + \varepsilon^2$

Must be cross-validated or badly biased



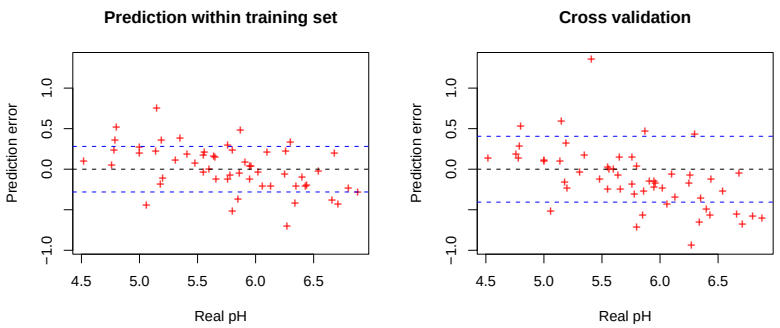
Note on slide 62. Root mean square error or RMSE is related to standard deviation, but it uses the known gradient values, and therefore the divisor is N instead of $N - 1$. Bias is defined as the mean of prediction errors, and the random error as the error about bias. We cannot use R functions `sd` and `var` directly, because these know only about unbiased estimates (divisor degrees of freedom $N - 1$), but we have to write out the equations:

```
error <- pH.wa3 - pH
rmse <- sqrt(sum(error^2)/length(error))
bias <- mean(error)          # Mean works OK
rander <- sum((error-bias)^2)/length(error)
```

Slide 63

Cross validation

Leave-one-out ('jackknife'), each in turn, or divide data into training and test data sets.



Note on slide 63. The simplest method of cross validation is to divide the data into training data set to develop the model, and to test data sets to apply and evaluate the model. This is very simple in R, but needs many commands. In the following, the first seven lines divide the data into two subsets, and remove the species that are missing in the training data sets. Three next lines develop the model with inverse regression in training data set, and finally, two last lines apply the model for the test data.

```
diatrain <- dia[1:55,]                      # Divide into two subsets
diatest <- dia[56:110,]
present <- apply(diatrain, 2, sum) > 0      # Take only species present in first
diatrain <- diatrain[,present]
diatest <- diatest[,present]
pH.train <- pH[1:55,]                       # Same division for pH
pH.test <- pH[56:110]
train.wa <- wascores(pH.train, diatrain)    # WA
train.pH <- wascores(train.wa, t(diatrain))
train.lm <- lm(pH.train ~ train.pH)         # Inverse regression
test.pH <- wascores(train.wa, t(diatest))   # Apply in second data set
test.pred <- predict(train.lm, data.frame(train.pH=test.pH))
```

With two data sets, the prediction error may be due to systematic differences between data sets in addition to random error [76].

Jackknife or leave-one-out is less wasteful to data, but takes more time, and needs a function to loop over observations, leaving each one out in turn, and evaluating the goodness of prediction for the removed observation. Current opinion is that leaving only one out changes the data too little and even this would exaggerate the goodness of prediction. K-fold cross validation leaves a certain proportion out in turn (1/5 for instance), and evaluates the goodness of model for each of these left out chunks in turn.

Slide 64

Bioindication: Likelihood approach

- **Likelihood** is the probability of a given *observed* value with a certain *expected* value
- Maximum likelihood estimation: Expected values that give the best likelihood for observations.
  - ML estimates are close to observed values, and the proximity is measured with the likelihood function
  - Commonly we use the negative logarithm of the likelihood, since combined probabilities may be very small

Note on slide 64. The presentation is based mainly on [76]. Similar ideas are elaborated by [35].

Note on slide 66. This is a straightforward application of the methods used in [76], but applying GAM instead of the Gaussian responses. The method is very general and can use any response functions, and even multiple gradients.

Comments on calculation to come (but it is very simple indeed).

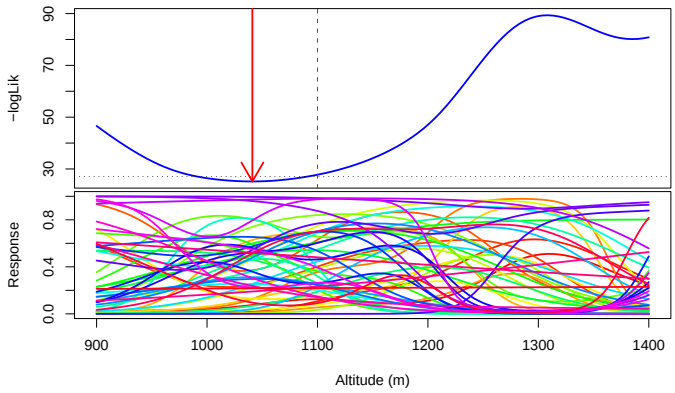
Slide 65

Regression and Bioindication

- **Regression:**
  - We know the gradient values x and observed species abundances y
  - We find the most likely expected values $\hat{\mu}$ for species
- **Bioindication:**
  - We know the observed species abundances y
  - We have a gradient model that gives the expected abundances μ for any gradient value x
  - We find the most likely gradient values $\hat{x}$ that maximize the likelihood of observing y when expecting μ
- ML Bioindication can be used with many response models and with many gradients

Slide 66

Finding elevation from species composition



3 Ordination

This chapter explains the basic exploratory ordination methods, or those methods which are based on the community composition only. Nowadays many vegetation ecologists prefer to use constrained methods where they try to express only that component of community variation that can be explained by the available environmental data. Those constrained methods are all special cases of the unconstrained methods of this chapter, and they will be discussed later (page 85).

Slide 67

Major ordination methods

- Principal Components Analysis (PCA)
  - Factor Analysis (FA)
  - Principal Co-ordinates Analysis (PCoA)
- Correspondence Analysis (CA)
  - Detrended Correspondence Analysis (DCA)
- Non-metric Multidimensional Scaling (NMDS)

Note on slide 67. The slide lists the most popular (or 'major') ordination methods. Commonly used multivariate methods are in the standard library `mva`. In R the major ordination methods can be applied using the following functions:

- PCA: `princomp` (library `mva`), `prcomp` (`mva`), `rda` (`vegan`)
  - FA: `factanal` (`mva`).
  - PCoA: `cmdscale` (`mva`).
- CA: `ca` (`multiv`), `decorana` (`vegan`), `cca` (`vegan`).
  - DCA: `decorana` (`vegan`).
- NMDS: `isoMDS` (`MASS`), help functions in `vegan`.

3.1 Principal components analysis

Note on slide 69. R library `mva` has two alternative versions of PCA: functions `prcomp` and `princomp`. Function `prcomp` is the recommended one: It uses singular value decomposition (slide 74) which is more reliable than the traditional algorithm in `princomp`. Function `princomp` is provided only for the S-plus compatibility.

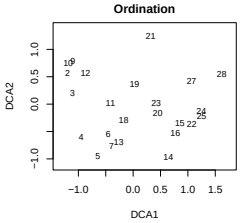
The usage is very simple: You just write

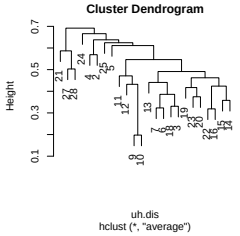
```
vare.pca <- prcomp(varespec)
```

Slide 68

Why ordination?

- Nobody should **want** to make ordination, but they are desperate with multivariate data.
- Show as much as possible with only a few axes, and explain those dimensions.






```
                  11111111122222222
         2345679012345689012345678
Cal.vul ++5.1..14185352.+.5.+...
Emp.nig 652+665737461676675746 74
Led.pal ..........+......54...42
Vac.myr ......1...1+24.4.76...67
Vac.vit 654566777766757678787776
Vac.uli .4+..6........4.2.4..+31
Dic.sp  .........1+.23..31+.87.1
Dic.fus +2113221114778414185.752
Dic.pol ...+.+111...+1.++5.+4..1
Hyl.spl ................4....+66
Ple.sch 3+4+++537666875774878899
Pol.jun +.+2+1111.+11++4+1.416.+
Pti.cil ++..+.1.3.1.+++1.6+4+1++
Cla.arb 267788236587768663665641
Cla.ran 887989568784678575566463
Cla.ste 99837199897+1+58+71++.4.
Cla.unc 113152114338531151436 3++
Cla.cor +1+1111111+21+11+++4122+
Cla.cri ++++1+111123311++114++++
Cet.niv .+61+131......++...+....
Ste.sp  .136+3.13.+23+21...4++.+
Cla.def +++11+11.124331232+42+++
```

Slide 69

Principal Components Analysis (PCA)

- Rotates the data in species space so that major part of the variation is shown on the first axes.
- Linear representation of the data: usually inadequate for community analysis but good for reducing environmental measurements.

where `varespec` is a vegetation data set in the `vegan` library. The default output is very voluminous. The function returns:

```
> str(vare.pca)
List of 3
 $ sdev    : num [1:24] 31.35 21.55 11.50  8.60  6.96 ...
 $ rotation: num [1:44, 1:24] -0.01399  0.01566 -0.00646 -0.05168  0.00858 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:44] "Cal.vul" "Emp.nig" "Led.pal" "Vac.myr" ...
  .. ..$ : chr [1:24] "PC1" "PC2" "PC3" "PC4" ...
 $ x       : num [1:24, 1:24] -10.8 -27.8 -25.7 -31.8 -19.6 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:24] "18" "15" "24" "27" ...
  .. ..$ : chr [1:24] "PC1" "PC2" "PC3" "PC4" ...
 - attr(*, "class")= chr "prcomp"
```

or three items:

- `sdev`: Square roots of eigenvalues based on species covariances.
- `rotation`: an orthonormal rotation matrix for all 44 species and for all 24 principal components. For a $n \times m$ matrix, the number of principal components is $\leq \min(m, n)$. Orthonormal means sum of squares for all species components is 1.
- `x`: PC scores for all 24 sites and for all 24 PCs. These are found by matrix multplication of the `rotation` and the data (`varespec`). The dispersion of these PCs depends on the eigenvalues and number of sites. The relation is in our example:

  ```
  > vare.pca$sdev[1:3]^2*(nrow(varespec)-1)
  [1] 22608.513 10678.993  3041.762
  > round(crossprod(vare.pca$x[,1:3]), 2)

           PC1      PC2     PC3
  PC1 22608.51     0.00    0.00
  PC2     0.00 10678.99    0.00
  PC3     0.00     0.00 3041.76
  ```

  which means that the dispersion of `x` is much larger than in `rotation`, and $n-1$ times the variance.

The alternative `mva` function `princomp` is very similar, but it uses divisor n for covariances instead of unbiased $n-1$ and so it reports a bit lower `sdev`. Further, it labels `rotation` as `loadings`, and `x` as `scores`, but the values are similar in both function (except possible sign changes).

Finally, `vegan` has a function `rda` for Redundancy Analysis, but it can be used for PCA as well:

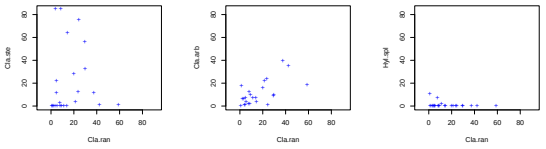
```
> rda(varespec)

Call:
rda(X = varespec)

              Inertia Rank
Total            1826
Unconstrained    1826   23
Inertia is variance

Eigenvalues for unconstrained axes:
   PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8
982.98 464.30 132.25  73.93  48.42  37.01  25.73  19.71
(Showed only 8 of all 23 unconstrained eigenvalues)
```

Function `rda` reports eigenvalues instead of their square roots (`sdev`), and these are based on unbiased covariances (divisor $n-1$). The `rda` result is very big, but there are several utility functions to access the results, such as `scores` to extract species or site scores, and `plot` for a graphical display. The scaling of species and site scores differs radically from `prcomp` and `princomp`, as will be discussed with slide 76.

Slide 70

Species space

- Graphical presentation of data matrix: Species are axes and span the space where sites are points.
- Some species show more of the configuration of sites than others.

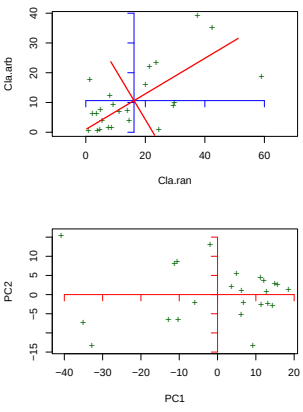



```
                  1111111112222222
         2345679012345689012345678
Cal.vul ++5.1..14185352.+.5.+...
Emp.nig 652+665737461676675746 74
Led.pal ..........+......54...42
Vac.myr ......1...1+24.4.76...67
Vac.vit 654566777766757678787776
Vac.uli .4+..6........4.2.4..+31
Dic.sp  .........1+.23..31+.87.1
Dic.fus +2113221114778414185.752
Dic.pol ...+.+111...+1.++5.+4..1
Hyl.spl ................4....+66
Ple.sch 3+4+++537666875774878899
Pol.jun +.+2+1111.+11++4+1.416.+
Pti.cil ++..+.1.3.1.+++1.6+4+1++
Cla.arb 2677882365877686636656 41
Cla.ran 8879895687846785755664 63
Cla.ste 99837199897+1+58+71++.4.
Cla.unc 1131521143385311514363++
Cla.cor +1+1111111+21+11+++4122+
Cla.cri ++++1+111123311++114++++
Cet.niv .+61+131......++...+....
Ste.sp  .136+3.13.+23+21...4++.+
Cla.def +++11+11.124331232+42+++
```

Slide 71

Rotation in species space

1. Move the axis origin to the centroid of the species space (species averages).
2. Rotate the axes so that the first axis becomes
   (a) As close to all observations as possible, which means that it
   (b) Explains as much of the variance as possible.
- First rotated axes show the configuration as well as possible.



Note on slide 71. Two-dimensional example data set is derived from `varespec` data set of the `vegan` library:

```
exa <- varespec[, c("Cla.ran","Cla.arb")]
exa.pca <- prcomp(exa)
```

PCA is a rotation, and function `prcomp` indeed labels the variable (species) solution as `rotation`:

```
> exa.pca
Standard deviations:
[1] 16.833855  7.166863

Rotation:
               PC1        PC2
Cla.ran -0.8583845  0.5130069
Cla.arb -0.5130069 -0.8583845
```

A rotation of angle θ in two dimensions can be made in matrix algebra by multiplying the target matrix with a rotation matrix [56]:

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{1}$$

The trigonometric functions in R use radians instead of the more familiar degrees (like *all* modern software does). The rotation for *Cladina rangiferina* seems to be:

```
> acos(diag(exa.pca$rotat))/(pi/2)*90
[1] 149.1357 149.1357
```

or 149°. The rotation is over 90° which means that the axis was reversed in `prcomp`: large values on axis 1 indicate small coverage of *Cladina rangiferina*. The "reversal" of an axis is quite a normal thing in PCA. It is important to remember that PCA is not defined for the sign reversal, but two identical solutions can be mirror image of each ohter. In this case, $149° - 90° = 59°$ rotation would give an identical solution.

The original axes were orthogonal, and so will be the axes after the PCA rotation:

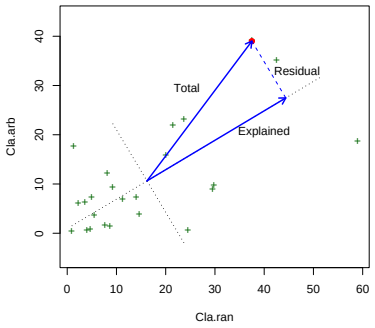
```
> asin(exa.pca$rotat[2,1])/(pi/2)*90
[1] -30.86432
```

We can ignore the sign, because axis reversals are the norm, and $59° + 31° = 90°$.

Slide 72

Explaining the variation

- **Total variation:** Sum of squared distances from the origin.
- **Explained variation:** Sum of squared projections onto principal components $= \lambda$.
- **Residual variation:** Sum of squared orthogonal distances from the principal components.
- Only a rotation: All $\min(S, N)$ axes explain everything.



Note on slide 72. Let's have a look at the `prcomp` result again:

```
> exa.pca
Standard deviations:
[1] 16.833855  7.166863
```

The default output gives the standard deviations of the axes. These are the square roots of the variances. The variances of the PCA axes are called eigenvalues. Because we made only a rotation, the variances of the axes or the sum of the eigenvalues equals the sum of variances in the data:

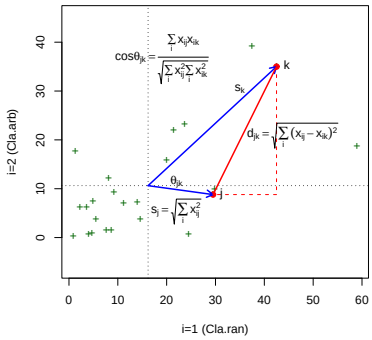
```
> sum(exa.pca$sdev^2)
[1] 334.7426
> sum(apply(exa, 2, var))
[1] 334.7426
```

Slide 73

How computer sees the configuration?

The configuration of points in species space can be defined either

1. As distances s_j, s_k and angles θ_{jk} from the origin.
   - Both included in covariances and variances.
   - $\cos\theta$ is Pearson's correlation coefficient.
2. As Euclidean distances d_{jk} between pairs of points.



Note on slide 73. Covariance can be expressed as $s_{jk} = s_j^2 + s_k^2 - 2\cos\theta s_j s_k$ [55].

Slide 74

Singular Value Decomposition (SVD)

$$x_{ij}^{(M)} = \bar{x}_j + \sum_{m=1}^{M} v_{im}\lambda_m^{1/2} u_{jm}$$

- An M–order least squares approximation of data matrix as a product of orthonormal loading matrices for sites v_{im} and species u_{jm} and corresponding eigenvalue λ_m
- Zero–order approximation: All species at their average abundace at the origin.
- When $M = \min(S, N)$ then approximation exact and $\sum \lambda =$ total variance.
- A numerical way of saying that PCA is only a rotation.

Note on slide 74. Function `prcomp` uses internally SVD (the algorithm is given with the slide 90). First the origin is moved to the centroid of the species space, and then the centred solution is decomposed:

```
> exa.cent <- scale(exa, scale=FALSE)
> exa.svd <- svd(exa.cent)
> range(exa.cent - exa.svd$u %*% diag(exa.svd$d) %*% t(exa.svd$v))
[1] -5.329071e-15  3.552714e-15
```

The last line makes matrix multiplication of the slide. The numerical accuracy seems to be pretty good in the SVD decomposition.

The PCA results are scaled a bit differently than in `svd`, but this will be discussed with the slide 75.

Slide 75

Loadings and scores

- Matrices **V** and **U** are orthonormal in SVD.
  - Average is zero $\sum_i u_{im} = 0$
  - Sum of squares is one $\sum_i u_{im}^2 = 1$
  - Components are uncorrelated $\sum_i u_{im}u_{in} = 0; \forall\, m \neq n$
- Normal scaling in PCA:
  - Singular values σ are scaled to covariance eigenvalues $\lambda = \sigma^2/(n-1)$.
  - Species loadings v_{jm} kept orthonormal: Rotation matrix.
  - Site scores scaled by singular values $\sigma_m^2 u_{im}$: Reflect the importance of axes and show the real configuration.
  - Sum of squares of scaled site scores $= \lambda_m(n-1)$.

Note on slide 75. Function `svd` (slide 74) returns the orthonormal matrices:

```
> crossprod(exa.svd$u)
             [,1]         [,2]
[1,]  1.00000e+00 -9.81906e-17
[2,] -9.81906e-17  1.00000e-00
> crossprod(exa.svd$v)
     [,1] [,2]
[1,]    1    0
[2,]    0    1
```

In contrast, the results of `prcomp` directly combine the eigenvalues λ with the scores: Species scores are orthonormal, but the site scores are scaled by the eigenvalues:

```
> crossprod(exa.pca$rotation)
    PC1 PC2
PC1   1   0
PC2   0   1
> round(crossprod(exa.pca$x), 2)

          PC1     PC2
  PC1 6517.71    0.00
  PC2    0.00 1181.37
```

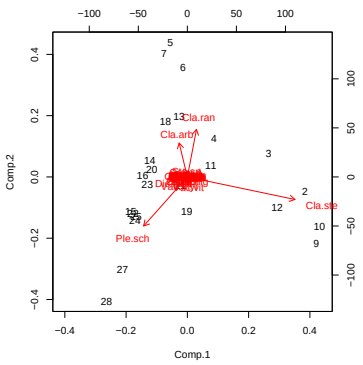
Here the diagonal elements give the raw crossproducts or $\lambda(n-1)$. Function `prcomp` reported square roots of eigenvalues (as `sdev`), and these eigenvalues λ were based on covariances. The crossproducts above are equal to $\lambda_k(n-1)$. After this combination of eigenvalues with site scores, the PCA rotation and scores directly approximate the data (cf. slide 74):

```
> range(exa.cent - exa.pca$x %*% t(exa.pca$rotation))
[1] -2.664535e-15  7.105427e-15
```

Slide 76

Biplot: Graphical SVD

- Species loadings and site scores in the same plot: Graphical order 2 approximation of the data.
- Origin: species averages.
- Species increases to the direction of the arrow, and decrease to the opposite direction.
- The longer the arrow, the stronger the increase.
- Approximation: Project site point onto species vector.



Note on slide 76. Although `prcomp` is the preferred function for PCA, the inferior `princomp` function has a canned `biplot` method to produce the biplot graph in the slide:

```
> biplot(princomp(varespec))
```

Please note that the biplot graph in the slide uses different axis scalings for species and sites. Although the biplot equations (slide 74) guarantee that the biplot approximates the data, it does not guarantee that species and sites are scaled similarly. Function `rda` in the `vegan` library tries to satisfy both desires: biplot approximation of the data, and similar scaling of species and site scores. The scaling is made by the proportional eigenvalue (eigenvalue divided by the sum of eigenvalues), and both species and sites are divided by the identical constant which guarantees that even after this proportional scaling the biplot approximation of the data holds.

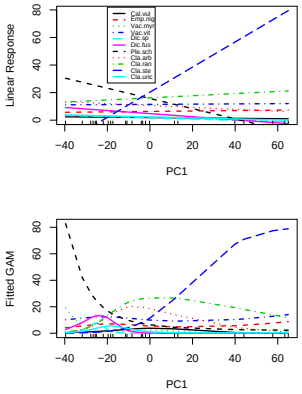
Note on slide 77. In the standard example, the loading of *Calluna vulgaris* is -0.01399:

```
> str(vare.pca$rotation)
 num [1:44, 1:24] -0.01399  0.01566 -0.00646 -0.05168  0.00858 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:44] "Cal.vul" "Emp.nig" "Led.pal" "Vac.myr" ...
  ..$ : chr [1:24] "PC1" "PC2" "PC3" "PC4" ...
> lm(Cal.vul ~ vare.pca$x[,1])

Call:
lm(formula = Cal.vul ~ vare.pca$x[, 1])

Coefficients:
    (Intercept)  vare.pca$x[, 1]
        1.87792         -0.01399
```

Slide 77

Linear response model

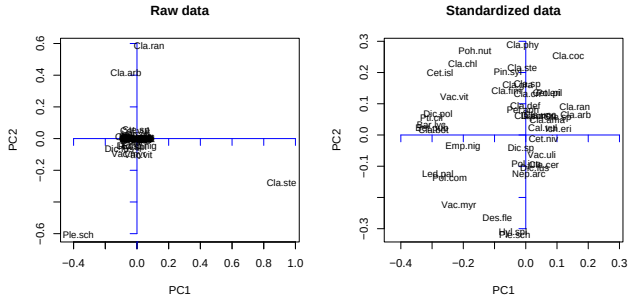
- PCA assumes that species have a linear regression with PCs.
- Loading is the regression coefficient between PC and the species.
- PCA may "fail": Real species responses rather unimodal despite PCA assumptions.



Slide 78

Standardized PCA

Standardizing species to unit variance (using correlation coefficient) makes all species equally important, instead of concentrating on the abundant species with largest variances.



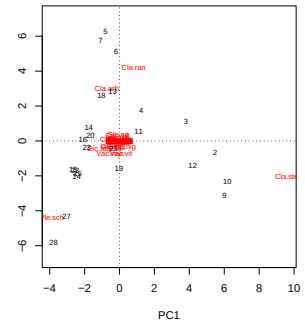
Note on slide 78. The standardized analyses can be done in the following ways with the alternative functions:

```
> prcomp(varespec, scale. = TRUE)
> princomp(varespec, cor = TRUE)
> rda(varespec, scale = TRUE)
```

Slide 79

PCA plot

- Axes must have identical scales.
- Origin is special and should be marked: Points near the origin are either average or poorly explained.
- Direction from the origin: The fastest change in species abundances.
- Distance from the origin: The magnitude of change.
- Angles between vectors: Similarity of the response.



Note on slide 79. The graph uses `plot` of function `rda` in `vegan`, whereas the previous graphs (mainly) were hand grafted from `prcomp`. The scaling of results is different in `rda` (see slide 76), so that both species and site scores can be nicely presented in a common diagram.

3.2 Factor Analysis (FA)

Factor Analysis is a very popular method in social and human sciences, and it is fairly conspicuous in several statistical packages. However, the implementation of FA is often poor, and sometimes a simple variant of PCA is called FA. The real FA is a Maximum Likelihood or other fancy methods instead of a rigid PCA rotation. It is best to use FA only in a confirmatory analysis where we have a specific hypothesis about the structure of the world (but, of course, we may have used FA to find out that hypothesis).

Note on slide 81. In R, FA is made by the function `factanal` (library `mva`) which uses Maximum Likelihood. The graph in the slide was derived with:

```
> data(varechem)
> factanal(varechem[,c("P","K","Ca","Mg")], 1)

Call:
factanal(x = varechem[, c("P", "K", "Ca", "Mg")], factors = 1)

Uniquenesses:
    P     K    Ca    Mg
0.340 0.399 0.163 0.310

Loadings:
```

Slide 80

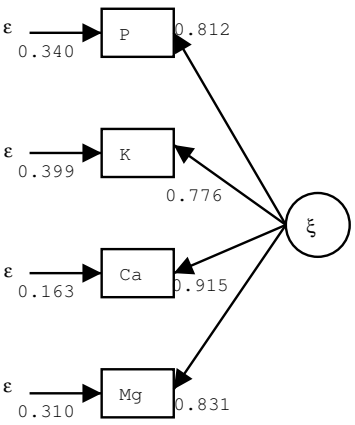
Factor Analysis (FA)

- Factor Analysis instead of PCA in many programs.
- Factor Analysis is a **statistical** method, whereas PCA is only a rotation.
  - World has two components: Common and unique.
  - FA tries to explain only the common component.
  - PCA discards later axes in hope of discarding unique variation, but that is mixed with all axes – kept and rejected.
- Latent variables ξ generate the observed variables x which have unique errors ε.
- Exploratory FA: Rotation to a simple structure.

Slide 81

Confirmatory Factor Analysis

- The real variables ξ are non-observable (latent), but they influence several observable variables x.
- Select a set of observable variables as indicators of the latent variable, and use them to build a *measurement model* for the latent variable.
- The latent variables are truer explanatory variables than single, noisy observed variables and can be used in further modelling.



```
   Factor1
P  0.812
K  0.776
Ca 0.915
Mg 0.831

                Factor1
SS loadings       2.789
Proportion Var    0.697

Test of the hypothesis that 1 factor is sufficient.
The chi square statistic is 4.93 on 2 degrees of freedom.
The p-value is 0.085
```

3.3 Principal Co-ordinates Analysis (PCoA)

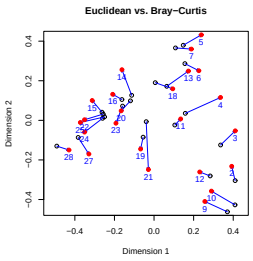
Principal co-ordinates analysis is also known as metric scaling or classic scaling. In R, it is made with function `cmdscale` (library `mva`). PCoA cannot be applied directly to a data matrix, but it needs a (dis)similarity matrix as an input. The (dis)similarities can be computed with function `dist` (`mva`). Libraries `vegan` (function `vegdist`) and `labdsv` (function `dsvdis`) provide some more alternatives for dissimilarities, among them those which are regarded as good choices in community ordination.

Slide 82

Principal Co-ordinates Analysis (PCoA)

A.K.A. metric or classic multidimensional scaling

- With Euclidean distances identical to PCA: Eigen analysis.
- Allows use of other dissimilarity measures: These may be mathematically naïve but ecologically elegant.
- Classical in vegetation ecology – still used in U.K.



Note on slide 82. See slide 73.

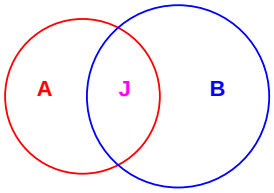
Note on slide 83. Dissimilarity indices are confusing for several reason. First, there are plenty of them [40, 54]. There is even a larger number of names of indices than there are indices, because many of these indices were invented independently numerous times. I have adopted a slack approach of using commonly known names in loose, generic meaning, but many textbook are more stringent and use different names, sometimes called 'correct' names. Different variants (quantitative and binary) of the Sørensen index of this slide are called Bray–Curtis, Czekanowski or Steinhaus index in the literature. If you use an index, you must back up your usage by giving an equation or a reference.

It is even more confusing that the notation used for indices varies a lot, and identical indices may look very different. I used set theoretical Venn diagrams above, because they

Slide 83

Dissimilarities for community data

- Presence/absence indices based on the number common species J on two sites compared to species richness A, B of sites.
- Similarity index s can be transformed to a dissimilarity index $d = 1 - s$.
- Quantitative generalizations:
  1. **Manhattan style**: use common part of abundance and sums.
  2. **Euclidean style**: use cross products and sums of squares.



Jaccard:	$\frac{J}{A+B-J}$
Sørensen:	$\frac{2J}{A+B}$
Ochiai:	$\frac{J}{\sqrt{AB}}$
Kulczyński:	$\frac{1}{2}\left(\frac{J}{A}+\frac{J}{B}\right)$

are graphical and intuitive. Most textbooks use a notation of 2×2 contingecy tables, where the species present in plot j are tabulated against species present in plot k

		Species in k	
		present	absent
Species in j	present	a	b
	absent	c	d

With this notation $J = a$, $A = a+b$ and $C = a+c$. Instead of similarities s, many textbooks (and even R and its functions) use directly dissimilarities. If these are defined as $1-s$ (other definitions can be used [54]), some common indices become:

Venn notation	2×2 notation	as dissimilarity
$\frac{J}{A+B-J}$	$\frac{a}{a+b+c}$	$\frac{b+c}{a+b+c}$
$\frac{2J}{A+B}$	$\frac{2a}{2a+b+c}$	$\frac{b+c}{2a+b+c}$

The contingency table has one entry that we have not used: d or the number of species missing in both sites. We have not used this, because we do not actually know how many species are missing in both sites: Obviously, most species of the world are absent from any two sample plots. Even most species of the regional or local species pool [25, 82, 111] are probably absent, and we should be able to single out the species that belong to the species pool of these two sites, but are absent. In the Venn diagram, these double absent species correspond to the area outside the circles, but we should decide how far away from the circles. However, some authors argue for using double misses, usually called double zeros. In most cases they define 'double zeros' as species that occur in the data set but not in these two plots. If we remove a sample plot with unique species, all dissimilarity indices among all remaining plots will change. It seems that d is an unknown quantity, and therefore no sensible index of community dissimilarity should use 'double zeros.'

Note on slide 84. The indices discussed here can be computed using function `vegdist` of `vegan`. Function `dsvdis` in Dave Roberts's `labdsv` package has all these same indices, and some more. The standard R funcion `dist` (recommended package `mva`) lacks many indices considered useful in community ecology. All return similar results which can usually be drop-in replacements for each other.

The nnomenclature is confusing in quantitative indices as well (cf. slide 83), but the notation seems to be a bit more consistent. However, if the index is expressed directly as a

Slide 84

The number of indices is a legio

Manhattan style		**Euclidean style**
$J = \sum_i \min(x_{ij}, x_{ik})$		$J = \sum_i x_{ij}x_{ik}$
$A, B = \sum_i x_{ij}$		$A, B = \sum_i x_{ij}^2$

	0/1	**Manhattan style**	**Euclidean style**
$\frac{J}{A+B-J}$	Jaccard	Jaccard	Similarity Ratio
$\frac{2J}{A+B}$	Sørensen	Steinhaus (Bray–Curtis)	Ružička
$\frac{J}{\sqrt{AB}}$	Ochiai	?	Cosine

dissimilarity instead of similarity, the indices may look fairly different:

$$1 - \frac{2\sum_i \min(x_{ij}, x_{ik})}{\sum_i x_{ij} + \sum_i x_{ik}} = \frac{2\sum_i |x_{ij} - x_{ik}|}{\sum_i x_{ij} + \sum_i x_{ik}}$$

Slide 85

Metric properties of indices

An index d is a **metric** if:

1. if $a = b$, then $d_{ab} = 0$ — minimum 0
2. if $a \neq b$, then $d_{ab} > 0$ — always positive
3. $d_{ab} = d_{ba}$ — symmetric
4. $d_{ac} + d_{cb} \geq d_{ab}$ — triangle inequality

Semimetric: Does not obey triangle inequality

Note on slide 85. Legendre and Legendre [54] discuss these properties in more detail. If a semimetric (or non-metric) index is used in PCoA, negative eigenvalues will be found on later axes.

Although it is customary to transform a similarity s into dissimilarity d with $d = 1 - s$, this is not a metric for Jaccard and Sørensen indices, e.g., whereas $d = \sqrt{1 - s}$ is a metric [54].

The Euclidean distance is a metric index. The relation between Euclidean distance and Cosine index was discussed with PCA (slide 73). Some of the popular distances are

3.4 Correspondence Analysis (CA)

CA is an eigenvector technique, just like PCA and PCoA. It replaces the Euclidean metric (slide 73) with Chi-square metric, and weights the sites and species by their totals in eigen analysis. The differences may sound small and technical, but in practice, the difference is huge. CA is a *much* better method for community ordination than similar PCA or PCoA. CA was introduced to ecologists by Mark Hill in 1973 [37], and since that it has been the most popular method of community ordination, in different variants.

There are numerous alternatives for running CA in R: Library `multiv` has function `ca`, library `vegan` functions `decorana` and `cca`. The latter are mainly for special variants of CA, but they run the basic CA as a special case. In addition, library `MASS` has function `corresp`, but that is intended for another use than community ordination.

The usage of these methods is simple:

```
> vare.ca1 <- ca(as.matrix(varespec))
> vare.ca2 <- decorana(varespec, ira=1)
> vare.ca3 <- cca(varespec)
```

Function `ca` does not have specific `print` or `plot` methods, but its elements must be accessed directly. However, functions `scores` and `ordiplot` in `vegan` can handle even `ca` results, and give plots directly (and, of course, they can handle `decorana` and `cca` results as well). Function `decorana` finds only four axes, but all $\min(m, n) - 1$ axes are found in `cca`.

Slide 86

Correspondence Analysis (CA)

Invented independently numerous times:

1. **Correspondence Analysis:** Weighted Principal Components with Chi-squared metric.
2. **Optimal Scaling:** Find site and species scores so that (i) all species occurring in one site are as similar as possible, but (ii) species of different sites are as different as possible, and (iii) sites are dispersed as widely as possible relative to species scores.
3. **Reciprocal Averaging:** Species scores are weighted averages of site scores, and simultaneously, site scores are weighted averages of species scores.

Note on slide 86. Hill [37] introduced CA as reciprocal averaging, and this name still sticks in vegetation science, although the algorithm is only rarely used any longer. Optimal scaling, also known as Dual scaling, is a popular approach in marketing research, social sciences and psychology [66], and it is the theoretical basis of the 'unimodal' properties of CA [87], further discussed in slide 92. The first approach is geometrical [30, 31], similar to PCA (see slide 73). Algorithmically it is the basis of modern software, although SVD is used in place of eigen analysis in `cca`, for instance.

Note on slide 89. The graph was produces with R function `assocplot`. The sites and the selected subset of most common species were arranged by the first CA axis (cf. slide 93). The graph indeed gives a graphical presentation of data matrix as CA sees it. The widths of the column are proportional to the sequare root of site totals or the weigths CA

Slide 87

Chi-squared metric

- *Metric* is a 'yardstick' to measure dissimilarities among points.
- PCA uses constant ('Euclidean') metric, but CA uses expected abundances as a metric.
- Expected abundances from marginal totals: Exactly like in χ^2 analysis of contingency tables.
- *Species profile* is the average proportion of species in the data, and *site profile* is the avarage proportion of sites.

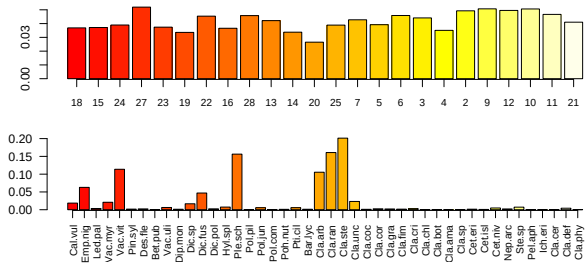
Euclidean: $d_{ij} = f_{ij} - e_{ij}$

Chi-squared: $\chi_{ij} = (f_{ij} - e_{ij})/\sqrt{e_{ij}}$

Slide 88

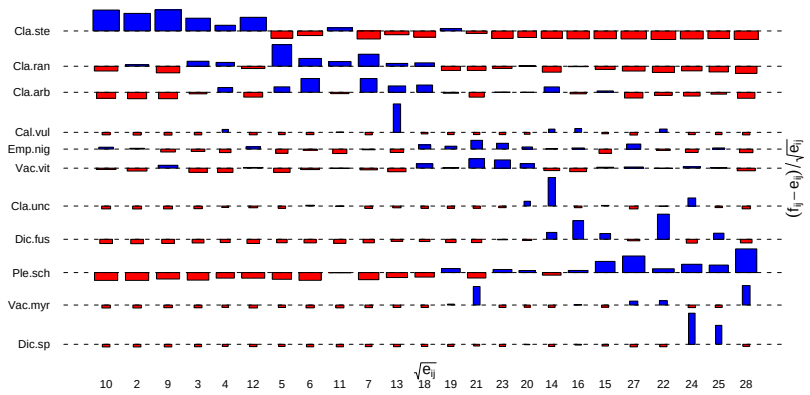
Species and site profiles

- All sites should have all species in the same proportions.
- Chi-squared distance is the difference between expected profile and real abundance distributions – both for the species and the sites.



Slide 89

Chi-squared transformation...

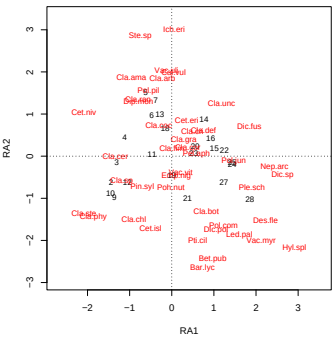


uses. The height of the column is proportional to the Chi-squared transformed data. So the surface area of the bar shows directly the relative importance of an observed value in CA.

Slide 90

...Weighted principal components rotation

- Weighted rotation in Chi-squared space:
  - Chi-squared transformation.
  - Marginal totals as weights.
- SVD in Chi-squared metric.
- Inbuilt double standardization for sites and species simultaneously.



Note on slide 90. To see how similar CA is to PCA, I give here simplified R code for both. PCA is just a wrapper for SVD (slide 74):

```
> PCA
function (X, scale=FALSE)
{
  X <- scale(X, scale=scale)                # step 1
  sol <- svd(X)                             # step 2
  sol
}
```

The data are centred, and optionally standardized to unit variance, in a step before SVD. The raw result is the object `sol` with three components: `d` is a vector of singular values, and `u` and `v` are orthonormal site and species score matrices. Normally these are still adjusted, so that singular values σ are transformed to eigenvalues by $\lambda = \sigma^2/(n-1)$ (see slide 75) and site scores are adjusted for eigenvalues.

CA is almost as simple [30]:

```
> CA
function (X)
{
  X <- X/sum(X)                                  # step 1
  rowsum <- rowSums(X)
  colsum <- colSums(X)
  rc <- outer(rowsum, colsum)
  X <- (X - rc)/sqrt(rc)
  sol <- svd(X)                                  # step 2
  sol$u <- sweep(sol$u, 1, 1/sqrt(rowsum))       # step 3
  sol$v <- sweep(sol$v, 1, 1/sqrt(colsum))
  sol
}
```

The preparatory step makes the Chi-squared transformation, then follows SVD, and finally, the weighting in PCA is taken into account at step 3. The object `sol` has again raw results of CA. Normally these are still adjussted so that $\lambda = \sigma^2$ and either species or sites scores (or even both symmetrically) are adjusted for eigenvalues.

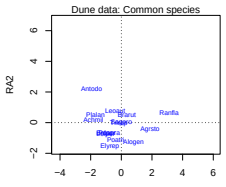
Slide 91

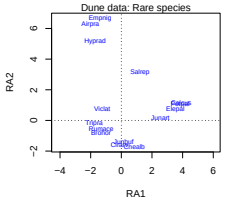
Rare species

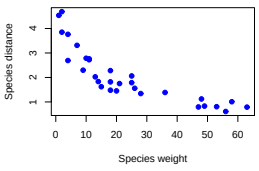
Rare species cannot have an average profile and are extreme in CA – but have small weights.

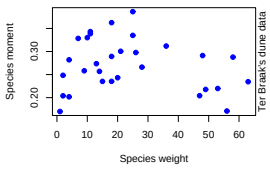
Solutions to the "problem":

1. Remove rare species.
2. Downweight rare species.
3. Do not show rare species: Popular.









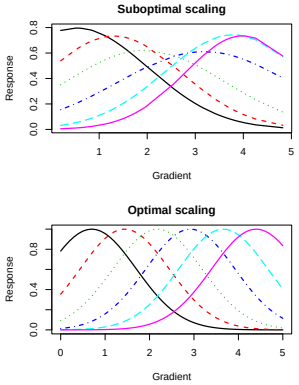
Note on slide 91. Rare species are usually extreme in ordination, and that has made people think that they are influential as well. Downweighting was introduced in the original DECORANA [38, 39], and it is available in the `vegan` port of `decorana`, and as an independent function `downweight`. Rune Økland [22, 69] advocates for downweighting, and even for more 'zealous' downweighting than standardly used.

Petr Šmilauer's plotting software for CANOCO [90] does not show "poorly fitted" species. Typically these are rare species, so *de facto* the "rare species problem" is solved by not displaying those species. It requires some diligence to find the option to turn on plotting all species.

Slide 92

When scaling is optimal?

- The species optima should be widespread: Measured by between species variance SS_B.
- The species responses should be narrow: Measured by within site variance of species optima SS_W.
- Total variance is $SS_T = SS_B + SS_W$.
- If most variance is between sites, the scaling is optimal.
- The criterion variable $\lambda = SS_B/SS_T$ is maximized in CA.



Note on slide 92. The presentation here is based on Nishisato's Optimal scaling [66]. Ter Braak reaches the same conclusion based on weighted averaging [87], see slide 95.

Slide 93

Ordered vegetation table

Raw

```
                  11111111122222222
         234567901234568901234578
Cal.vul ++5.1..14185352.+.5.+...
Emp.nig 652+665737461676675746 74
Led.pal ..........+......54...42
Vac.myr ......1...1+24.4.76...67
Vac.vit 654566777766757678787776
Vac.uli .4+..6........4.2.4..+31
Dic.sp  .........1+.23..31+.87.1
Dic.fus +2113221114778414185.752
Dic.pol ...+.+111...+1.++5.+4..1
Hyl.spl ................4....+66
Ple.sch 3+4+++537666875774878899
Pol.jun +.+2+1111.+11++4+1.416.+
Pti.cil ++..+.1.3.1.+++1.6+4+1++
Cla.arb 2677882365877686636656 41
Cla.ran 8879895687846785755664 63
Cla.ste 99837199897+1+58+71++.4.
Cla.unc 1131521143385311514363++
Cla.cor +1+1111111+21+11+++4122+
Cla.cri ++++1+1111233111++114++++
Cet.niv .+61+131......++...+....
Ste.sp  .136+3.13.+23+21...4++.+
Cla.def +++11+11.124331232+42+++
```

PCA

```
          2212221212 1 12 11  1  1
          875245634078531691432290
Ple.sch 9988887767+5+64+774+6353
Dic.fus 2578.7857424141311121+21
Cla.arb 1476566676887838667652 23
Vac.myr 7626..4.+....17.4.....1.
Dic.sp  1.2+873..3...+1.....1...
Cla.unc ++5463338521131514313111
Hyl.spl 66...+...4..............
Cal.vul ..35+.5.5+.2.8.1.45+1+.1
Pol.jun +.1.16+41+1+2+1+41+..+11
Vac.uli 13.4.+...264......+4....
Led.pal 24.4.........+5.........
Ste.sp  +.3.+++42.326+.+1331...1
Cla.def ++3+2+3443+112212.++1+11
Pti.cil +++++1+4...+.16+13.+.+1.
Cla.cri ++31++143++1+211+1++1+11
Dic.pol 1.+.4.1+.++.+.5.+1....11
Cla.cor +21+12+42+111++111+11+11
Cet.niv .......+..1+1..++.6+..31
Vac.vit 67777758676756866745767 7
Emp.nig 471546676667+47663257657
Cla.ran 3665647647989858587878 56
Cla.ste .411+.++++15377788899999
```

CA

```
       1    1  1 11122211122222
       029342561738913046572458
Cet.niv 1.3+6.1+.1.++.+.........
Cla.ste 99998937817587++++141+..
Cla.ran 685877988988556747665643
Ste.sp  1..13.6+33+21.4.2+3..+++
Cla.arb 3226757868886676746561
Vac.uli ...4+....6.4...2...34.+1
Cal.vul 1+.+51.14.82...+553.5+..
Emp.nig 765527+63647677666175464
Vac.vit 76754756766768876577 7776
Cla.cor 1+11+11111+11+4+2+12+12+
Cla.cri 1+1++1+11+21+14+313+1+++
Pti.cil .+1+...+3.1+164..+++++1+
Cla.def 1+1++111.+212243433++2++
Dic.pol 1.1...+.1+..+5++.1+..4.1
Cla.unc 1111331542311135835+463+
Pol.jun 1+1.+.2+11++414+1+1..16+
Led.pal .........+..5.....44..2
Dic.fus 1+2211131244115478758.72
Ple.sch 335+46++7+65747767898889
Vac.myr ..1.......1.47..+4266..7
Dic.sp  .....1....+..1.3.32.+871
Hyl.spl ..............4...6..+6
```

Note on slide 93. Nice, diagonal ordering is regarded as ideal for vegetation tables, and it is achieved by CA due to its optimal scaling properties (slide 92). This ordering is possible with one dominant gradient only. For this reason, CA is used a "seriation" method in archaeology and other fields where data should be arranged in a sequence.

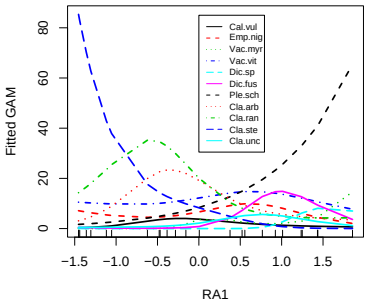
Function `vegemite` in `vegan` can automatically arrange the vegetation table according to the first CA (or PCA etc.) axis, if its result is given in the parameter `use`, and it can also change the original percent cover values into one-digit cover codes:

```
> vegemite(varespec, use=cca(varespec), scale="Hult")
```

Slide 94

Unimodal response

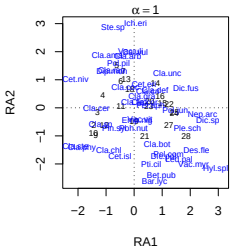
- Optimal scaling tries to pack species occurrences into tight parcels: Unimodal response model.
- Eigenvalue λ tells the success of packing – but too high a value ($\sim$ 1) indicates disjunct subsets of sites.

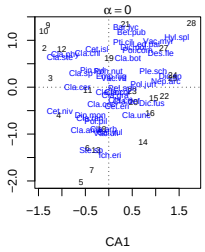


Slide 95

Reciprocal weighted avarages

- Repeated application of weighted averaging leads into CA.
- Weighted averages shrink by factor λ and need restoring.
- Species scores can be weighted averages of site scores ($\alpha = 1$), or vice versa ($\alpha = 0$), or restored symmetrically ($\alpha = \frac{1}{2}$).





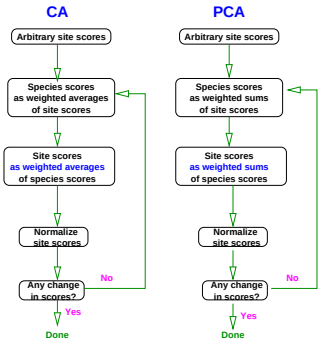
$$u_i = \lambda^{\alpha-1} \frac{\sum_j x_{ij} v_j}{\sum_j x_{ij}}$$

$$v_j = \lambda^{-\alpha} \frac{\sum_i x_{ij} u_i}{\sum_i x_{ij}}$$

Slide 96

Power algorithm

Arbitrary vector multiplied with high power of a matrix converges to the eigenvector of the matrix.



- Repeated application of weighted averaging gives CA solution.
- Repeated application of weighted sums gives PCA solution.
- Both methods simple, but numerically poor.
- In fact, CA and PCA both are best made using [weighted] SVD of [Chi-square transformed] matrix.

Note on slide 94. The curves are fitted GAMs.

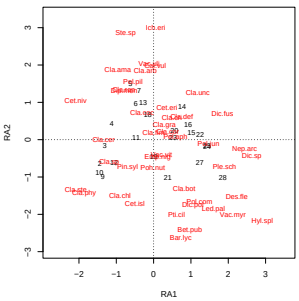
Note on slide 96. When CA was introduced as Reciprocal averaging [37], it was seen as intuitively clearer and simpler method than PCA, and this was one reason for its success. PCA was seen as a computer magic compared to simple and intuitive elegance of RA. However, PCA can be done with similar "weighted summation", although nobody does so, because the algorithm is poor: unreliable and slow. Although CA is still presented as a RA method, all modern software use better, but more complicated and less intuitivea lgorithms. The recommended algorithm uses SVD (slide 90).

Slide 97

CA: Joint plots

Biplot with weighted Chi-squared metric: Species and sites in the same plot (but scaling α counts).

- Distance from the origin: Chi-squared difference from the profile.
- Points at the origin either average or poorly explained.
- Distant species often rare, close species common.
- Unimodal interpretation: Species optima and gradient values – at least for well explained species.



Note on slide 97. The joint plot problems are further discussed in [71].

Slide 98

Eigenvalue in CA

1. The factor of shrinking in weighted averaging: $\lambda = 0 \ldots 1$.
2. The proportion between site variation from total variation: $\lambda = 0 \ldots 1$.

- Sum of all eigenvalues = mean squared contingency coefficient: The statistic decomposed into linear components in CA.
  - No ecological meaning: "Proportion of 'variance' explained" neither meaningful.
  - Variance is never evaluated in CA.
- Eigenvalues close to 1 suspious: Probably a disjunct subset of points.

3.5 Detrended Correspondence Analysis (DCA)

Detrending Correspondence Analysis was introduced by Mark Hill in 1980 [38, 39] when he published the programme DECORANA. Ordinary CA had been the method of choice earlier, and DCA was to improve upon that method. It became soon the method of choice. However, in mid-1980s criticism was growing [60, 72, 98]. This criticism was not the reason of demise of DCA, but it was replaced by a new improvement upon CA: Canonical Correspondence Analysis (CCA) [88, 89, 90] which has been the method of choice since then. DCA has still some advocates, who think that it is the best method available [67, 69].

It is very difficult to make a distinction between a *method* of DCA and a piece of software for DCA, because all implementations of DCA use the same code base [38]. If there was a bug in the original code base, the bug may remain in various, unrelated programmes [78]. The function `decorana` in the `vegan` library is a straightforward port of the central analytic engine of DECORANA. The usage is simple:

```
> decorana(varespec)

Call:
decorana(veg = varespec)

Detrended correspondence analysis, with 26 segments.
Rescaling of axes with 4 iterations.

               DCA1   DCA2   DCA3    DCA4
Eigenvalues  0.5249 0.1572 0.0967 0.06075
Axis lengths 2.8161 2.2054 1.5465 1.64864
```

The function has `plot`, `summary` and `scores` methods.

Note on slide 99. Downweighting was already discussed in slide 91.

Note on slide 100. The regular, one-dimensional response in the graph is known as the *species packing model*, already discussed in slide 24 [29, 99, 101]. The species packing model was not seen only as a simplistic device for vegetation simulation and modelling, but rather as a theoretically expected shape. The same packing model is central even in the current standard model [87, 94].

Slide 99

Detrended Correspondence Analysis (DCA)

Mission to correct three 'artefacts' in CA:

1. **Detrending** to remove 'spurious' curvature in ordination.
2. **Rescaling** to correct shrinking at the ends of ordination axes.
3. **Downweighting** to reduce the influence of rare species.

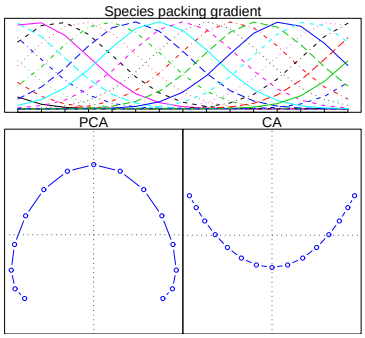
Program DECORANA the only implementation – difficult to separate a program from the method.

Slide 100

Detrending CA: The argument

PCA and CA both produce a curve from a single, ideal gradient, but the shapes have one important difference:

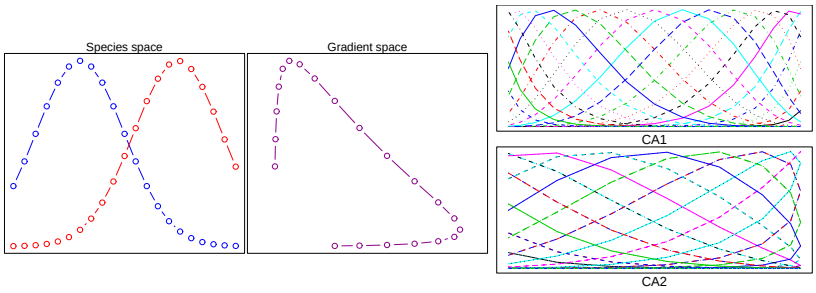
- Horseshoe in PCA curved inwards at ends: Wrong order along the first axis.
- Arc in CA preserves the correct ordering along the first axis: Worth of detrending.



Slide 101

The birth of the curve

- There is a curve in the species space, and PCA shows it correctly.
- CA may be able to deal with unimodal responses, but if there is one dominant gradient, the second optimal scaling is folded first axis.

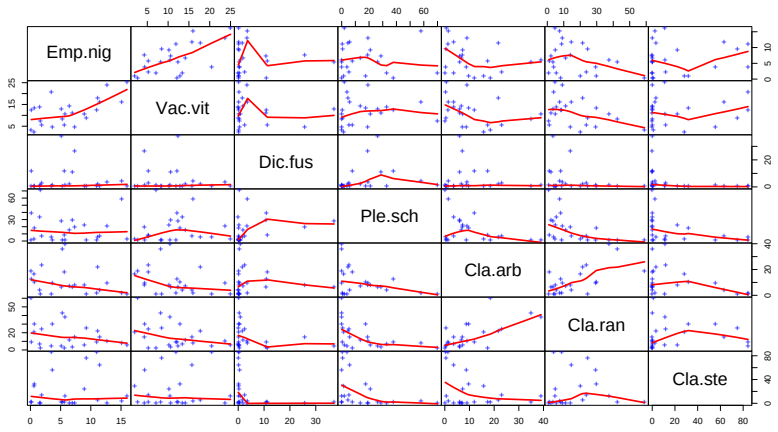


Note on slide 101. The graphs are wrongly labeled (but I am too lazy to re-draw the graph): swap *Species space* and *Gradient space*.

This is another example of one-dimensional fixation in DCA (the first was on the slide 100). If the data are strongly one-dimensional so that the first 'gradient' is at least two times longer than the second one, then folding can cause the curve in CA [44, 94].

Slide 102

But is there a curve in species space?



Note on slide 102. The `pairs` plot for the most abundant species:

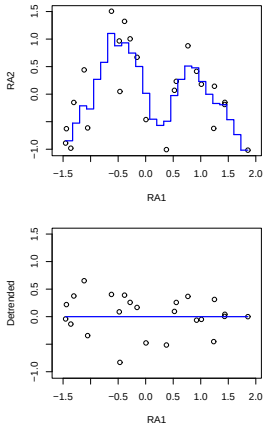
```
> tot <- colSums(varespec)
> pairs(varespec[, tot>100], panel=panel.smooth)
```

Note on slide 103. DCA was criticized for too zealous detrending: It will remove all structure, not only the undesired curvature. This seems to happen in the `varespec` dataset

Slide 103

Detrending by segments

- Divide the axis into segments (default 26).
- Average using moving, weighted windows in segments and take the residuals as the new 2nd axis.
- Ideally the direction of 2nd axis changes to an important, non-linearly independent gradient.
- May remove information as well, not only the arc like desired.

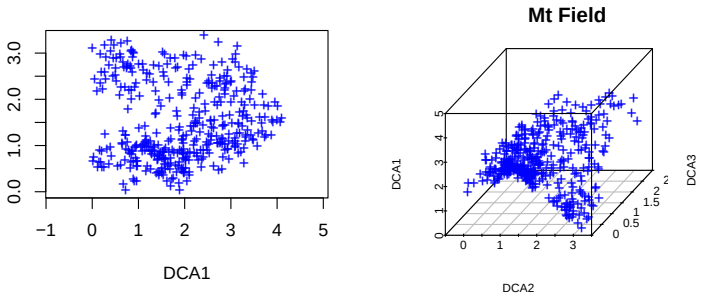


in the slide, where detrending picks one deviant plot (number 21) instead of removing only the arc effect. Ter Braak suggested an alternative of polynomial detrending which should remove only the curve, but leave other structure undamaged [89, 90]. However, comparative studies revealed that the polynomial detrending was even more dangerous than detrending by segments, and it should not be used [51].

Slide 104

Detrending artefacts

- DCA configuration usually a triangle or a diamond: an artefact.
- DCA detrends twisting the space: 'Lollypop' or 'lasagna' effect.



Note on slide 104. DCA ordination can be usually identified as a DCA without reading the legends, because the points form a triangle or diamond shaped pattern. When the DCA result is inspected in 3D or spinned on a screen, it becomes evident that detrending works by twisting the space so that empty space is left in the middle of points. This cannot be seen in 2D plots, but even there, one end of the first axis (typically the upper end) has a wider range along axis 2. This does not mean that there would be larger variation in the vegetation at that end, but the twisting exposes that variation. On the other hand, the

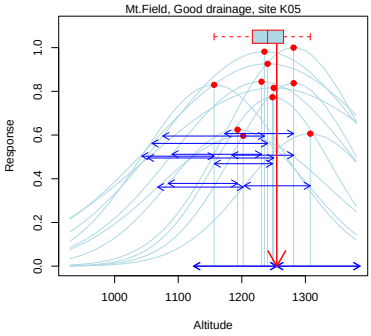
variation at the opposite end is twisted into axis 3, which should be inspected for this end.

Slide 105

Hill indices of beta diversity

1. Average width of species responses.
2. Variance of optima of species occurring in one site.

- Used with scaling of ordination axes.
- The first index discussed and described, but the second applied.
- Equal only to degenerated species packing gradients.



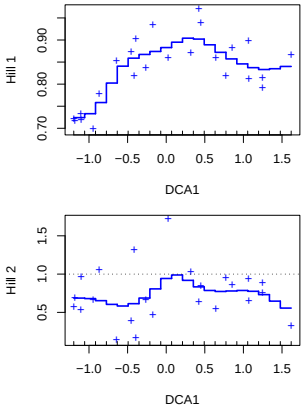
Gaussian responses fitted to species occurring in one site.

Note on slide 105. This is a duplicate of slide 53.

Slide 106

Rescaling and downweighting

- Rescale gradients so that the weighted variance of species scores = 1 along the axes.
- Often said that average width of responses is 1 'sd' unit, but that is never done.
- Downweighting: Reduce abundances of rare species so that they have still less weight on ordination.



Note on slide 106. Most articles, beginning from the original introduction of DCA [39], say that rescaling is based on the widths of species responses and the axes are scaled into 'sd' units. Hill mentions briefly in the DECORANA manual [38] that this is not done, but weighted variances of species scores are used. Oksanen and Tonteri [80] discuss this in more detail, and demonstrate that the two alternative rescaling criteria may be in conflict. The issue was already discussed in slide 55.

Experimental package `gravy` from my web pages contains function `hillscale` for assessing *both* indices of beta diversity, and for rescaling observed gradients by weighted variances of species scores.

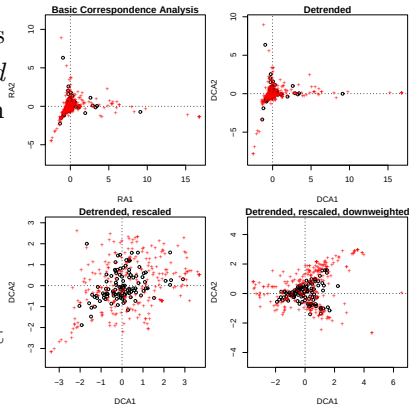
Slide 107

Method or Programme?

It seems that DECORANA rather is a piece of software than a *method* of ordination, since it is a collectin of three different tricks:

1. Detrending
2. Rescaling
3. Downweighting

Rescaling often is more important than detrending...

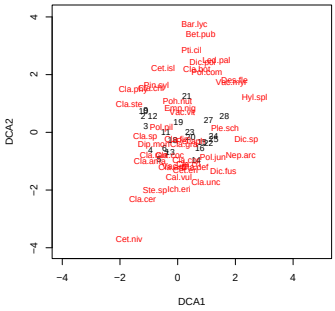


Slide 108

DCA plot

DCA plots often taken more seriously than other ordinations:

- Axes taken as gradients, and even scaled in ecologically meaningful 'sd' units.
- Species scores taken as species optima.
- Species scaled so that site scores are their direct weighted averages ($\alpha = 1$).



Note on slide 108. It is customary to scale DCA so that the smallest site score is 0. Then the largest site score gives directly the 'axis length'. Species scores have a wider dispersion than site scores in DCA (the wider, the lower is the eigenvalue), and some species scores will be negative. Function `decorana` in the `vegan` library does not follow this convention, but it uses the real origin (weighted average of site or species scores), although the axis lengths are similar as in other DCA software.

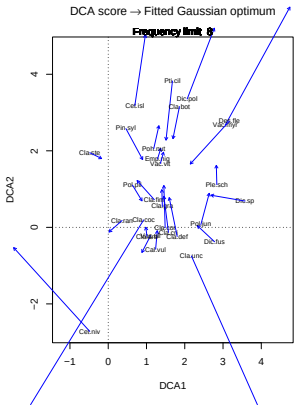
Slide 109

Is DCA based on Gaussian response model?

Cajo ter Braak:

> "Four conditions (equal tolerances, equal or independent maxima, and equally-spaced or uniformly distributed optima and sample points) are needed to show that (detrended) correspondence analysis provides an *approximate* solution to the unimodal models." *Biometrics* **41,** p. 870 (1985).

= (D)CA can *approximate* Gaussian model – *if* we have infinite species packing gradients.



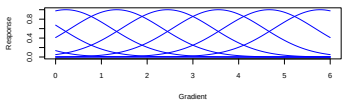
Note on slide 109. The optima were found fitting two-dimensional GLM (slide 39), but forcing the width to $t = 1$ (as it is assumed to be in DCA). Only a couple species had a two-dimensional Gaussian response without fixing t. These same conditions [87] were discussed in slide 110.

Slide 110

Weighted averages are good estimates . . .

. . . of **species optima** if:

1. Sites x are evenly distributed about optimum u
2. Sites are close to each other



. . . of **gradient values** if:

1. Species optima u evenly distributed about site x
2. All species have equal response widths t
3. All species have equal maximum abundance h
4. Optima u are close to each other

Conditions true only for infinite species packing gradients.

Note on slide 110. Based on Cajo ter Braak's analysis [87], although the formulation differs from the original.

3.6 Non-metric Multidimensional Scaling (NMDS)

NMDS was the major challenger of DCA in the 1980s, and it is still regarded as the best and most robust unconstrained ordination method. Its reputation was based on its performance as a test winner in simulations [60]. However, it never was popular. One important reason may be that DCA (and later CCA) came with a really good and readily available software, but no such software was available for NMDS. Another reason is that the use of DCA was simple and direct, but numerous choices — like the choice of dissimilarity index — had to be made in NMDS. Further, NMDS is an iterative method with no guaranteed convergence. This means that the user must inspect several alternative results and someway decide whether she has reached a good solution, and to recognize that solution. In 1980s computers were slow and limited in memory (1 MB being a huge memory capacity). NMDS was much more magic than easy and direct DCA, which became the standard tool for a long time.

NMDS is a strong method mainly because of its robustness [60]. Robustness became a universal catch word after Minchin's simulation studies [60, 61], and it was used for any nice method, including DCA. However, the meaning of *robustness* is very explicit and clear: It is the ability of a method to work *when assumptions are violated* (slide 153). DCA was justified by a specific model (see slide 100), but it was not robust against deviations from that model [60], quite the contrary. NMDS was able to deal with *any* kind of response model.

There were several programmes for NMDS, but about the only one to work by the recommended procedure was Peter Minchin's DECODA [62]. DECODA is still available, and it is the recommended choice. However, library `vegan` complements other R functions so that a very DECODA like analysis can be performed in R. The proper NMDS module is function `isoMDS` in the `MASS` library, but `vegan` provides better [26] dissimilarity indices (`vegdist`), common data standardization methods (`decostand`), comparison of alternative dissimilarity indices and transformations (`rankindex`), random starts (`initMDS`), scaling and rotation of results (`postMDS`), comparison of results (`procrustes`), and an opportunity to add species scores (`wascores`) or fit environmental variables (`envfit`, `ordisurf`) to the ordination.

Slide 111

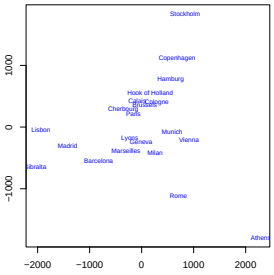
Non-metric Multidimensional Scaling (NMDS)

- Rank-order relation with community dissimilarity and ordination distance: No specified form of regression, but the the best shape is found from the data.
- Non-linear regression can cope with non-linear species responses of various shapes: Not dependent on Gaussian model.
- Iterative solution: No guarantee of convergence.
- Must be solved separately for each number of dimensions: A lower dimensional solutions is not a subset of a higher, but each case is solved individually.
- A test winner.

Slide 112

MDS is a map

- MDS tries to draw a map using distance data.
- MDS tries to find an underlying configuration from dissimilarities.
- Metric MDS (PCoA) assumes linear relation, but NMDS finds any relation.
- Only the configuration counts:
  - No origin, but only the constellations.
  - No axes or natural directions, but only a framework for points.

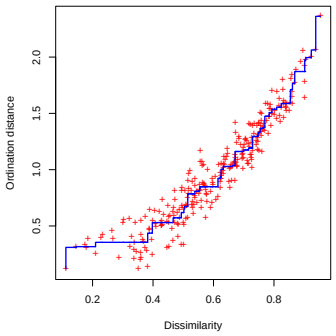


Map of Europe from road distances.

Slide 113

Monotone regression

- Measured community dissimilarities and ordination distances have similar rank ordering.
- No specified shape, but can cope with different response shapes.
- Sum of squared residuals from the regression: *Stress*.
- The model behind: Finds gradients if dissimilarities meaningful.
- Iterative solution: No guaranteed convergence.



Note on slide 113. There are several variants of NMDS, but the description here applies to Kruskal's original NMDS (which was found the best choice for community ordination [60]).

In Kruskal's NMDS, the stress S is defined as [97]:

$$S^2 = \frac{\sum_{i \neq j} [\theta(d_{ij}) - \tilde{d}_{ij}]^2}{\sum_{i \neq j} \tilde{d}_{ij}^2}$$

where d are the observed dissimilarities, $\tilde{d}$ the distances in ordination space, and θ refers to the monotonic regression function. Function `isoMDS` raports $100S$, or "per cents".

There are other NMDS algorithms. Many software packages use ALSCAL or alternating least squares which is closely related to the original Kruskal method, but performed more poorly in tests [**?**]. Another method used in R is the Sammon scaling (function `sammon` in `MASS`) which defines a linear, weighted stress function [97]:

$$S = \frac{1}{\sum_{i \neq j} d_{ij}} \sum_{i \neq j} \frac{(d_{ij} - \tilde{d}_{ij})^2}{d_{ij}}$$

Because of the first multiplier, this weights small dissimilarities, and may be able to find the structure in the local neighbourhood of points rather than the global gradient structure. Typically the points are evenly dispersed over the ordination spaced, almost like in a grid. The convergence of `sammon` is much more difficult than in `isoMDS`, and a very large number of random starts is needed for any kind of stable solution. In its neighbourhood emphasis the method differs strongly from the metric scaling as well, and `cmdscale` gives typically very bad starting solutions.

Sammon scaling belongs to a group of local neighbourhood methods which are currently very hip. Among these are Self-Organizing Maps (SOM), also known as Kohonen maps and their approximation, Principal Curves [20]. There are two R packages for SOM: `GeneSOM` and `MASS` which has functions `SOM` and `batchSOM`. There are two packages for Principal Curves as well, `pcurve` and `princurve`, both based on Glenn De'Ath's code [20]. Both need much care in application, and at least `SOM` does not easily fulfill its promises of local scaling, but tends to be rather linear. Someday I will write a whole new chapter about these methods...

Slide 114

Recommended procedure

1. Use adequate dissimilarity indices: An adequate index gives a good rank-order relation between community dissimilarity and gradient distance.
2. No convergence guaranteed: Start with several random starts and inspect those with lowest stress.
3. Satisfied only if minimum stress configurations are similar.

Note on slide 114. The recommended procedure is based on Minchin's extensive simulation work [60]. Although this work is effectively published, most software does not follow the recommendations — except Minchin's DECODA [62].

The points will be expanded and the recommended procedure in `vegan` explained in slides 115 to 117.

Slide 115

Good dissimilarity measures for gradients

- The Model: Find a a dissimilarity measure that describes correctly gradient separation.
- Bray–Curtis (Steinhaus), Jaccard and Kulczyński.
- 'Wisconsin double standardization' often helpful: First standardize each species to equal maximum, then sites to equal totals.
- Euclidean distance and Chi-squared dissimilarity are poor.

Note on slide 115. We have already discussed the dissimilarity indices in slides 83 and 84. In NMDS the situation is actually simpler than it appeared earlier: The most important property of an index is its *rank order similarity to gradient separation* [26]. This is more important than the 'metric properties'. Moreover, several different indices may be rank-order similar, so it does not matter which of them is used. Further, after some standardization even more indices may become rank-order similar. For instance, Manhattan, Kulczyński and Bray–Curtis dissimilarities are rank-order identical after the Wisconsin double standardization.

Package `vegan` provides the recommended dissimilarity indices [26] (`vegdist`), such as Bray–Curtis and Kulczyński indices. Library `labdsv` provides the same, and some more indices (`dsvdis`). In addition, `vegan` provides some standardization methods for vegetation data (`decostand`), among them the Wisconsin double standardization (`wisconsin`) which often improves clearly the analysis. In addition, function `rankindex` can be used to find the rank correlation (cf. slides 113, 115) between dissimilarities and gradient separation. The following procedure compares the indices against the environmental data (all environmental variables are `scale`d to have standard deviation 1):

```
> rankindex(scale(varechem), varespec)
      euc       man       gow       can       bra       kul
0.15162055 0.17849802 0.07625823 0.17465086 0.18919631 0.18982872
> rankindex(scale(varechem), wisconsin(varespec))
     euc       man       gow       can       bra       kul
0.2812121 0.2869565 0.1962582 0.1568379 0.2869565 0.2869565
> rankindex(scale(varechem), sqrt(varespec))
     euc       man       gow       can       bra       kul
0.1878788 0.1748617 0.1023979 0.1620553 0.2161792 0.2134914
```

Square root transformation and the Wisconsin double standardization both improve the results, and the best choice (among these alternatives) seems to be use Bray–Curtis (or Kulczyński or Manhattan) index and Wisconsin double standardization.

Note on slide 116. Many programs start NMDS from a solution of metric scaling (slide 82), in R from a solution of `cmdscale` (library `mva`). So does R (`isoMDS`, library `MASS`).

Slide 116

Starting MDS

- Most software packages start with metric MDS (PCoA), but this is dangerous:
  1. Metric MDS may be close to a local minimum which traps the iteration.
  2. Hides the uncertainty of convergence from the user (and that is unmoral).
- Start with random configurations, and compare minimum stress solutions.

This is dangerous, because metric MDS is likely to be close to some local optimum. For finding the global optimum, several random starts must be applied. DECODA runs several random starts, and compares them all agains each other. Library `vegan` does not provide this functionality, but a simpler procedure may be followed: Save the best solution so far, and continue as long as you can make some improvement. Please note that the default convergence criterion (`tol`) is pretty slack in `isoMDS`. Here are some steps that can be followed

```
> vare.dist <- vegdist(wisconsin(varespec), "bray")
> mds.null <- isoMDS(vare.dist, tol = 1e-07)
> mds.1 <- isoMDS(vare.dist, initMDS(vare.dist), maxit = 200,
        trace = FALSE, tol = 1e-07)
# If mds.1 has a lower stress than mds.null, do the following
# and continue:
> mds.null <- mds.1
```

Note on slide 117. Procrustes rotation (`procrustes`) is the only reliable way of seeing that two configurations are identical. Of course, you can study only the solutions with lowest stress. However, mere similar low stress is not a sufficient guarantee of identical solutions: In particular in large data sets some single points can be at very different locations with almost identical low stress. DECODA uses numerical criteria and decision rules to say which solutions are identical. In `vegan` it is better to rely on visual inspection of Procrustes diagrams:

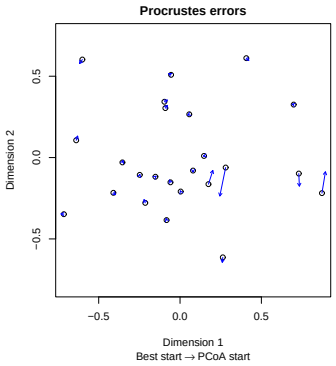
```
# After previous steps, two best solutions are saved in mds.1 and mds.null
> plot(procrustes(mds.1, mds.null))
```

Note on slide 118. The example data are 1132 sample plots by North-East passage in Northern Russia and Siberia. The plot shows the location of one species-poor plot in Kolyma. This plot shares species with two other sites only, and so it could just as well be on either side of the axis connecting these two sites. Step-across dissimilarities (slide 158) are sometimes useful in these cases. Other ordination methods are prone to similar outliers, although the symptoms are different. For instance, correspondence analysis tends to polarize outlier site against all other sites with eigenvalue close to 1: most sites are lumped together, and the deviant sites are alone at the far end of the axis.

Slide 117

Comparing configurations: Procrustes rotation

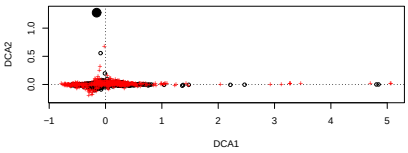
- Procrustes rotation to maximal similarity between two configurations:
  - Translate the origin.
  - Rotate the axes.
  - Deflate or inflate the axis scale.
- Single points can move a lot, although the stress is fairly constant: Especially in large data sets.

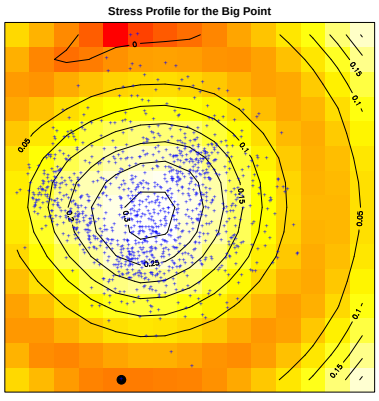


Slide 118

Outliers in the outskirts

- Points that have just a little in common with just a couple of sites are difficult to locate
- Often a marginal circle
- Sometimes a point could be in two alternative locations

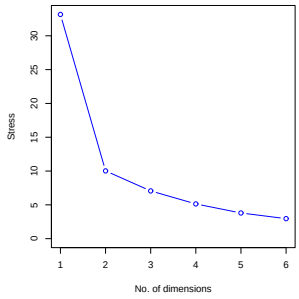




Slide 119

Number of dimensions

- In eigenvector methods axes are orthogonal and previous axes remain unchanged when new axes are evaluated.
- MDS Solutions for each number of dimensions separate.
- Adequate number of dimensions difficult to know: After sudden drop of stress is a good idea.



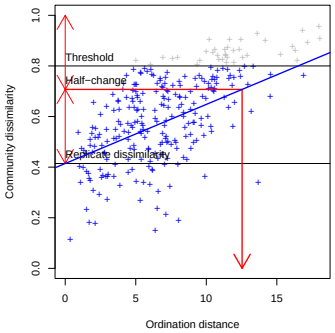
Note on slide 119. PCA is a projection and a rotation (slide 71), and so is CA (slide 90), but NMDS is a *mapping* (slide 112). The best mapping is dependent on the dimensionality of mapping. For the best mapping, we should know the *real* dimensionality of the data — and for that we have to assume that data has *the real* dimensionality. Unfortunately, there are no really good ways of doing this. The *sree plot* of this slide is the most commonly used method. Unfortunately the scree plot is often difficult to interpret. A joking rule of thumb – often attributed to Kruskal – is that most data sets have 2.5 dimensions: two that you can plot on a flat paper, and something extra that you must explain in the text. This vagueness of dimensionality is a reason why some people dislike NMDS: they want easy and clear solutions.

Slide 120

Scaling of axes

MDS axes have no unique scaling or direction, but all rotations and scalings are equally good solutions.

- Customary to rotate to principal components: First dimension most important.
- Half-change scaling give ecologically meaningful units.



Note on slide 120. Map is a map even if you turn it round – and Australians indeed like to astonish us with maps where south is upward. Since the original NMDS program (KYST),

it is customary to rotate solutions to principal components. This indeed only a rotation and it does not influence the *configuration* of points in any way. However, it makes the 'longest' dimension horizontal, and two solutions easier to compare to each other.

Another indeterminacy in NMDS is that the axes do not have any natural scaling: You can multiply all coordinates with some factor, and the *configuration* is unchanged. Minchin's DECODA introduces a half-change scaling wo that one unit corresponds to halving of community similarity.

PC rotation and half-change scaling can be both performed with function `postMDS` in `vegan`. For half-change scaling it needs the original dissimilarity matrix as well:

```
mds.1 <- postMDS(mds.1, vare.dist)
```

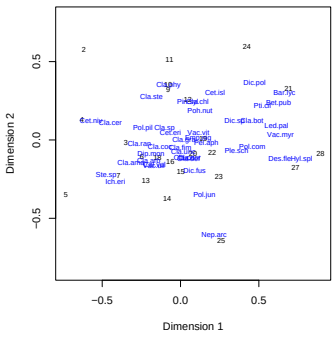
Another alternative is to rotate a NMDS configuration so that the first axis is maximally similar to a defined environmental variable. In `vegan` this can be achieved with function `procrustes` using an environmental variable as the target. Function `fitted.procrustes` returns (as default) results in the units of the target. The following rotates the `mds.1` configuration to Humus depth (`Humdepth`) and plots the results using its units (cm):

```
> mds.humus <- procrustes(Humdepth, mds.1)
Warning message:
X has fewer axes than Y: X adjusted to comform Y
 in: procrustes(Humdepth, mds.1)
> ordiplot(fitted(mds.humus))
Warning message:
Species scores not available in: ordiplot(fitted(mds.humus))
```

Slide 121

MDS plot

- All that counts is the configuration: Axes are nothing.
- No origin, but just a map.
- Species can be added as their weighted averages.



Note on slide 121. Function `wascores` in `vegan` can be used to find species scores for a NMDS solution.

4 Ordination and environmental variables

This chapter is badly in need of upgrading

The basic ordination methods of the previous chapter all were based on the community composition only. They did not take into account the environmental information in any way. In modern times, it is much more popular to use the environmental information directly and use constrained or 'canonical' ordination methods [44, 54, 81, 88, 89, 90, 94]. The unconstrained analyses should be still used, and indeed, they should often be used in place of constrained analysis [67]. Constrained analysis is well suited for confirmatory research, where we have specific *a priori* hypotheses on the important variables, and we want to test those hypotheses. In exploratory analysis we just want to get an idea of environmental variables that might be important. In those cases it is better to use unconstrained analysis with environmental information. Constrained analysis does not try to show all variation in the data, but it tries to extract only that component of variation that can be explained by the constraints. It may happen that we have no observations on some important variables, and this remains unnoticed if we look at the constrained ordination only [45].

Slide 122

Ordination and environment

We take granted that vegetation is controlled by environment, so

1. Two sites close to each other in ordination have similar vegetation, and
2. If two sites have similar vegetation, they have similar environment; moreover
3. Two sites far away from each other in ordination have dissimilar vegetation, and perhaps
4. If two sites have different vegetation, they have different environment

4.1 Interpreting ordination

Interpreting means usually explanation environmental terms. The most popular method is to fit vectors onto an ordination (slide 123) after the example of CCA (slide 133). However, the assumptions of slide 122 guarantee that close sites are similar in environment, but they do not guarantee a linear change in the environment accross compositional space. Therefore fitted surfaces are more reliable (slide 124) — even in CCA.

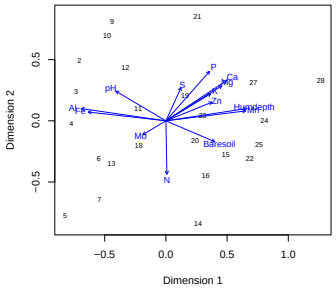
Note on slide 123. Library `vegan` has a function `envfit` to fit environmental variables onto an ordination. It can make a distinction between continuous variables and fit vectors for them (call function `vectorfit`) and factors (class variables), and fit weighted averages for them (call function `factorfit`). It can also add vectors into an ordination plot, and it test the 'significance' of fitted vectors or factors with permutation tests.

Note on slide 124. Library `vegan` has a function `ordisurf` which can fit smooth surfaces of environmental variables onto ordination. The function uses `gam` (library `mgcv`) with thin-plate splines to fit the smooth surface, and selectes the degree of smoothing with generalized

Slide 123

Fitted vectors

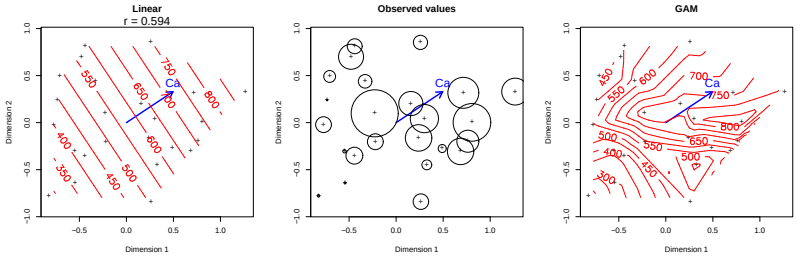
- **Direction** of fitted vector shows the gradient, **length** shows its importance.
- For every arrow, there is an equally long arrow into opposite direction: Decreasing direction of the gradient.
- Implies a linear model: Project sample plots onto the vector for expected value.
- Class values as weighted averages.



Slide 124

Alternatives to vectors

- Fitted vectors natural in constrained ordination, since these have linear constraints.
- Distant sites are different, but may be different in various ways: Environmental variables may have a non-linear relation to ordination.

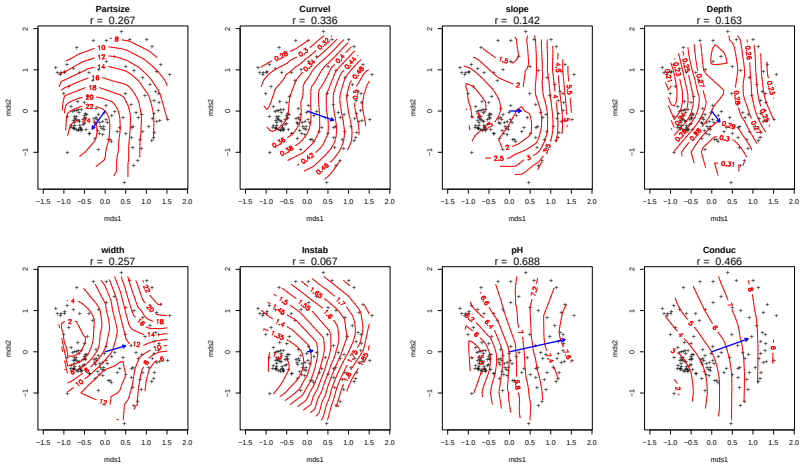


cross-validation [107]. With thin-plate splines, the surfaces have the same smoothness in all directions. Then the function uses `interp` (library `akima`) to transfer the fitted values into a regular grid for plotting.

The middle graph uses function `symbols` which can be used to plot circles, squares etc. of given size at any location in the plot.

Slide 125

Example: River bryophytes



Note on slide 125. Data courtesy of Risto Virtanen and Timo Muotka.

Slide 126

Lessons from environmental interpretation

- Environmental variables need not be parallel to ordination axes.
- Axes cannot be taken as gradients, but gradients are oblique to axes: You cannot tear off an axis from an ordination.
- **Never** calculate a correlation between an axis and an environmental variable.
- Environmental variables need not be linearly correlated with the ordination, but locations in ordination can be exceptional.

4.2 Constrained ordination

The most important constrained ordination methods are Redundancy analysis (RDA) and Canonical correspondence analysis (CCA), both invented by Cajo ter Braak [88]. RDA is based on the PCA framework and it is purely linear method. CCA is based on the CA,

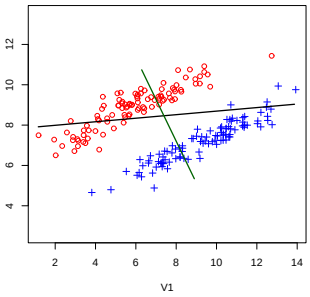
and it has a better ability to handle unimodal responses. Cajo ter Braak wrote a computer programme CANOCO which became the most popular ordinatin programme in community ecology, and replaced Mark Hill's DECORANA [38]. CANOCO uses the same code base, and there is continuum both in the methods and in the users of these programmes. Latest versions of CANOCO have a Windows GUI and advanced support software for plotting and further analysis.

CCA and RDA are the two most popular ordination methods. Their properties are described in detail in many good summaries [44, 54, 81, 88, 94], and there are many good practical introductions into their use [49, 90].

Slide 127

Constrained vs. unconstrained aims

- Unconstrained ordination tries to display the variation in data.
- Constrained ordination tries to display only the variation that can be explained with constraining variables.
- You can observe only things that you have measured.



Note on slide 127. The graph is based on Anderson (**ref**). Økland [67] has a further discussion on the niches of constrained and unconstrained ordinations.

Slide 128

The constraining toolbox

- **Linear tools** based on PCA framework:
  - Discriminant analysis, Canonical Correlations
  - Redundancy Analysis (RDA).
  - Only RDA useful in community ecology – if linear model is adequate.
- **Unimodal tools** based on CA framework:
  - Constrained or 'Canonical' Correspondence Analysis (CCA).
  - Absolutely the most important constrained ordination in ecology: The only one dealt with in these lectures.

Note on slide 128. R functions `cca` and `rda` provide fairly similar functionality as basic CANOCO. The functions use `formula` interface:

```
cca(varespec ~ Humdepth + Baresoil, varechem)
cca(varespec ~ Humpdeth*Baresoil, varechem)      # Interactions
cca(varespec ~ . , varechem)                     # All of varechem
rda(dune ~ Moisture, dune.env)                   # Factor constraint
                                                 # ... with ouput (print)
Call:
rda(formula = dune ~ Moisture, data = dune.env)

              Inertia Rank
Total           84.12
Constrained     27.49    3
Unconstrained   56.64   16
Inertia is variance

Eigenvalues for constrained axes:
  RDA1   RDA2   RDA3
19.067  5.764  2.656

Eigenvalues for unconstrained axes:
    PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10
15.0767  9.4719  6.6546  5.2645  4.9862  3.5484  3.1803  2.4353  1.7665  1.5447
   PC11    PC12    PC13    PC14    PC15    PC16
 0.9967  0.6459  0.4060  0.2962  0.1968  0.1664
```

The functions have `summary`, `plot` and `scores` methods. Further supporting functions will be discussed with the following slides.

In addition, Stéphan Dray has a dedicated library `CoCoAn` with function `CAIV` for CCA, called as "Correspondence Analysis with Instrumental Variables". `CAIV` is still used in most graphs in these lectures, because I had not written `cca` yet when I made the first pass.

The classical statistical models of Discriminant analysis and Canonical correlations differ from the corresponding constrained model of RDA. For instance, Canonical correlations are indeed canonical which means, among other things, that the analysis is symmetric and you can change environmental and community data without changing the result. In RDA (and CCA) the analysis is non-symmetrical so that you constrain the community data by the environmental data.

In R, Discriminant analysis can be performed with function `lda` or linear discriminat analysis, and there is also function `qda` for quadratic discriminant analysis, both in `MASS`. Canonical correlations are in function `cancor` (library `mva`).

Note on slide 130. The SVD algorithm of RDA is similar to the SVD algorithm of PCA (discussed with slide 90), but with one more step. All species individually are first regressed on the constraints, and the fitted values of this linear reggression are submitted SVD. CCA is similar, but it uses site totals as weights in the linear regression and then submits the fitted values to CA through SVD (slide 90).

Note on slide 131. The graphics is based on Mike Palmer [81], but the original algorithms are naturally based on ter Braak [44, 88]. Essentially, CCA is similar to the power algorithm in CA (slide 96), but with two alternating steps: (1) regression from constraints to LC scores, and weighted averaging from LC to WA scores.

The preferred algorithm is based on SVD [54] of slide 130, but the alternating regression algorithm of the slide is usually represented — sometimes given almost as a definition of CCA. Function `cca` in `vegan` uses SVD. Moreover Stéphane Dray's `CAIV` uses an eigenvector algorithm which also is superiour to the algorithm of the slide.

Slide 129

Constrained Correspondence Analysis (CCA)

Ordinary Correspondence analysis gives:

1. **Site scores** which may be regarded as describing the gradients.
2. **Species scores** which may be taken as location of species optima in the space spanned by site scores.

Constrained or 'Canonical' Correspondence Analysis gives in addition:

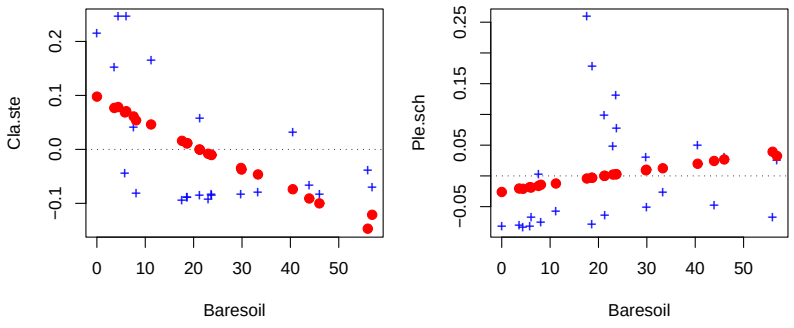
3. **Environmental scores** which define the gradient space.

And optimizes the interpretability of results.

Slide 130

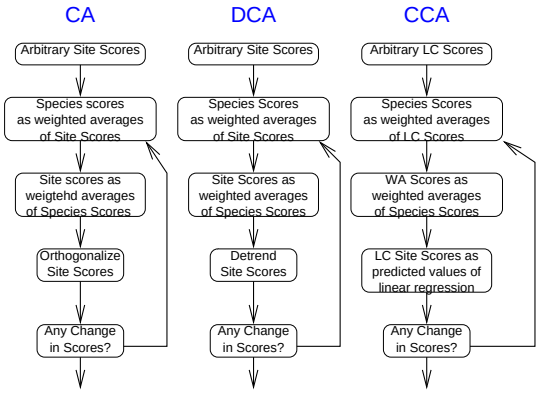
CCA: Algorithm

1. Fit weighted linear regression to all species individually using all constraints as explonatory variables.
2. Analyse fitted values using CA



Slide 131

CCA: Alternating regression algorithm



Two kind of site scores:

1. **LC scores** are predicted values of multiple regression with constraining variables = the constraints.
2. **WA Scores** are weighted averages of species scores.

Slide 132

Those numbers...

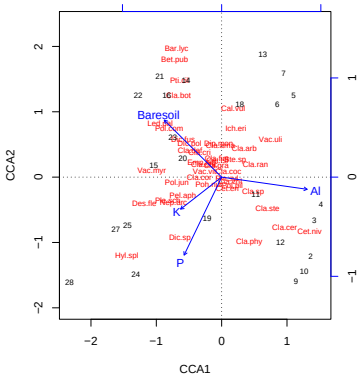
- Eigenvalues exactly like in CA.
  - CCA eigenvalue *should* be lower than in CA – or constraining may have been useless.
  - Eigenvalue has nothing to do with variance, so there is neither 'variance explained'.
- Species – Environment correlation: Multiple correlation from constraining regression: Usually high even with poor models.
- Pointwise goodness of fit can be expressed either as residual distance from the ordination space or as proportion of projection from thet total Chi-squared distance – exactly like in PCA.

Note on slide 132. Species – environment correlation can be strongly influenced by single, deviant points — just like usual correlation. Therefore latter axes with lower eigenvalues can have surprisingly high correlations.

Slide 133

CCA plot

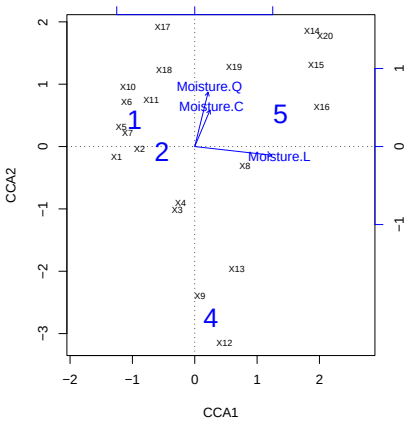
- Like CA plot, but now a **triplot**: Vectors for *linear* constraints.
- Classes as weighted averages.
- Most use LC scores: These are the constraints.
- Popular to scale species relative to eigenvalues, but keep sites unscaled.
- Sites do not display the configuration, but their projections onto environmental vectors are the estimated values.



Slide 134

Class constraints

- Class variables usually as 'dummy' variates: Make $m-1$ indicator variables out of m levels
- Indicator scoring: 1 if site belongs the the class, 0 otherwise
- One dummy less than levels, because all are redundant
- Ordered factors may be better expressed with polynomial constraints



Note on slide 134. There is one less dummy variables than levels because we know that a site *must* belong to some class: It it does not belong to any of $m-1$ one classes, it must belong to the remaining. All information on the class memberships can be expressed with $m-1$ dummy variables. CANOCO drops the last class, but R (like most statistical software) drops the first class. This is called as corner point contrasts.

The user does not need to split the class variables into 'dummy' variables, because R knows factors. The Dutch dune enviroment data [44] in the `vegan` package looks this:

```
> summary(dune.env)
```

```
       A1          Moisture Management       Use     Manure
 Min.   : 2.800   1:7      BF:3       Hayfield:7   0:6
 1st Qu.: 3.500   2:4      HF:5       Haypastu:8   1:3
 Median : 4.200   4:2      NM:6       Pasture :5   2:4
 Mean   : 4.850   5:7      SF:6                    3:4
 3rd Qu.: 5.725                                    4:3
 Max.   :11.500
```

Only `A1` (thickess of A_1 horizon in cm) is a continuous variable, and other variables are factors. R changes these internally into dummy variables without bothering the user:

```
> model.matrix(~ Management)
   (Intercept) ManagementHF ManagementNM ManagementSF
1            1            0            0            0
2            1            0            0            1
3            1            0            0            1
...
19           1            0            0            1
20           1            1            0            0
attr(,"assign")
[1] 0 1 1 1
attr(,"contrasts")
attr(,"contrasts")$Management
[1] "contr.treatment"
```

The `(Intercept)` which is all ones is not used in the analysis. R knows several contrast types for changing factors into model matrix, and the corner point contrast is known as `contr.treatment`. Users can defined other type of constraints, but these do not necessarily make sense in the ordination diagrams. However, ordered factors are defined as polynomial constraints. A four-level ordered factor would be expressed in three 'dummy' variables: linear, quadratic and cubic effects:

```
 > model.matrix(~ Moisture)
   (Intercept) Moisture.L Moisture.Q Moisture.C
1            1 -0.6708204        0.5 -0.2236068
2            1  0.6708204        0.5  0.2236068
3            1 -0.2236068       -0.5  0.6708204
...
19           1  0.2236068       -0.5 -0.6708204
20           1 -0.6708204        0.5 -0.2236068
attr(,"assign")
[1] 0 1 1 1
attr(,"contrasts")
attr(,"contrasts")$Moisture
[1] "contr.poly"
```

This may be a useful contrast in ordination diagrams, because it helps in finding the cases where we could replace a multilevel factor with a single continous variable.

Package `vegan` uses always the dummy variables (or `model.matrix`) in displaying the factor effects. It would be desirable to have class centroids, and they will be added in the future versions of `vegan`.

Note on slide 135. The diagram uses LC scores, but still the major variable becomes involuted. In principle, the LC scores are the exact (scaled and weighted) environmental variables, and the projections are exact, but only in the full m-dimensional space, and the straigh lines are curved when projected onto a plane. With WA scores the best fitted surfaces and projections diverge in any case.

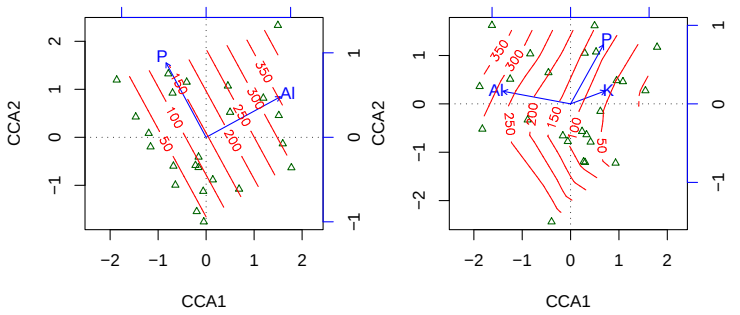
The diagram was produced with:

```
> plot(tmp <- cca(varespec ~ Al + P + K), dis=c("bp","lc"), type="p")
> ordisurf(scores(tmp, dis="lc"), Al, add=T)
```

Slide 135

Predicted values of constraints

- Project a site point onto environmental arrow: Prediction
- Exact with two constraints: Multidimensional space warped



Slide 136

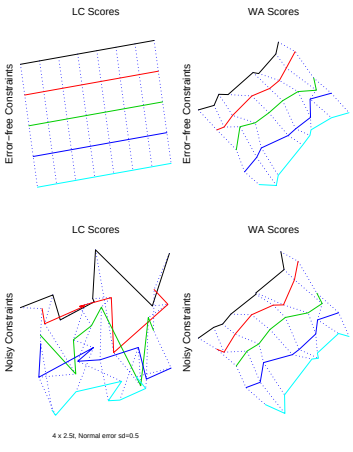
LC or WA Scores?

Mike Palmer:

- Use LC scores, because they give the best fit with the environment, and WA scores are a step from CCA towards CA.

Bruce McCune:

- LC scores are excellent, if you have no error in constraining variables. Even with small error, LC scores become miserable, but WA scores are good even in noisy data.



Note on slide 136. Mike Palmer [81] made people aware of the existence of two kind of CCA scores, and recommended the use of LC scores. In early publications only WA scores were used, and they were the only available scores in CANOCO version 2. Later, Petr Šmilauer used LC scores as defaults in his CANODRAW, like Mike Palmer found out [81]. Recently the LC scores are used nearly always, and Cajo ter Braak has endorsed this practice [**?**].

McCune's [58] criticism has been often misunderstood: He did not warn us against the use of *noisy* environmental variables, but against the use of LC scores: All observations are 'noisy', and LC scores are sensitive to this natural variation in the data. In contrast, WA scores seem to be rather reliable even with natural variation in the environmental variables.

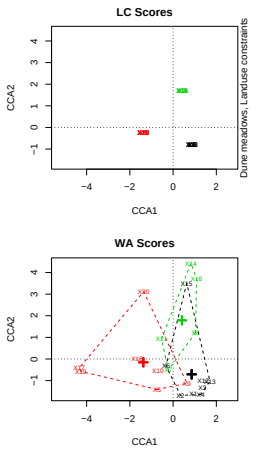
Functions `rda` and `cca` in `vegan` use WA scores in plots as a default, but either (or both) are available on request. Function `CAIV` (library `CoCoAn`) knows only LC scores.

Slide 137

WA and LC scores with class constraints

1. Make class centroids as distinct as possible.
2. Make clouds about centroids as compact as possible.

- Success $\approx \lambda$.
- LC scores are the class centroids: The expected locations.
- If high λ, WA scores are close to LC scores.
- With several class variables, or together with continuous variables, the simple structure becomess blurred.



Note on slide 137. McCune [58] in the slide 136 was practical: WA scores should be used, because they work better than LC scores. There are also more principled arguments: WA and LC describe different things, and depending on needs, either can be used. Often the scientist would want to get the WA scores and is surprised when seeing the LC scores that were not wanted.

Most commonly this happens with factor constraints, and is a frequently asked question in the ORDNEWS mailing list. LC scores are linear combinations of environmental variables, and factors have only a limited number of possible values. This means that all observations with identical environmental conditions, all belonging to the same class, will have identical LC scores. With a single factor, the number of different plotting positions is the number factor levels (classes). It does not matter how different the vegetation is within factor levels, but they are still plotted over each other. The LC scores with continuous constraints are similarly only the (scale) environmental variables, and the ordination configuration is equivalent to plotting sites with using environmental variables as axes. In CCA, the equivalence is slightly distorted by weighting, but in RDA the equivalence is exact. This is not as clearly visible with continuous variables as it is with classes, which therefore are the greatest source of surprise. With large number of environmental variables, the LC are still the environmental variables, but this becomes obscured due to multidimensionality.

WA scores are found from the community composition. If the community composition differs, the WA scores are different even if the environmental variables are identical. If the environment controls the composition, the WA scores will be very near to the corresponding

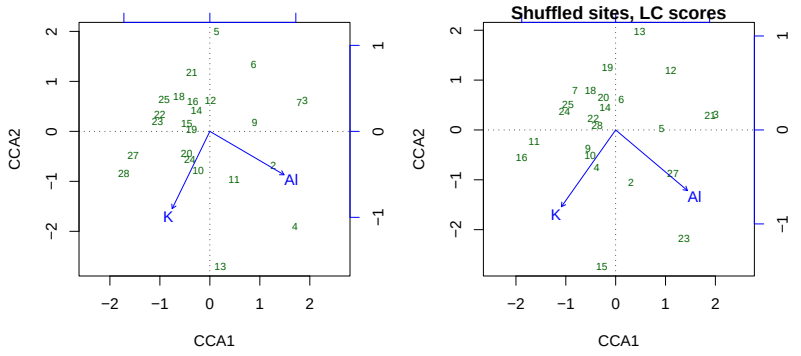
LC score. With poor variables, the difference is large. The LC scores are the constraints and the WA scores are the constrained community ordination. Therefore `vegan` uses WA scores as the default.

Library `vegan` has a small utility function `spider.cca` which plots both the LC and WA scores in the same graph and combines these with a line. In models constrained by a single factor the result is a 'spider' where each WA score is combined to its class centroid. These plots can be used to asses the goodness of constraining variables. This usage is similar as in the classical Discriminant analysis [56, 97]: the LC scores give the class centroids, and the discriminant function tries to assign observations as close to these centroids as possible.

Slide 138

LC scores are the constraints

LC scores *are* the scaled and weighted constraints and shuffling the order of sites of the community data does not change the configuration.



Note on slide 138. The graph was produced using:

```
> plot(orig <- cca(varespec ~ Al + K, varechem), dis=c("lc","bp"))
> i <- sample(nrow(varechem))
> plot(shuff <- cca(varespec ~ Al + K, varechem[i,]), dis=c("lc","bp"))
> mtext("Shuffled sites, LC Scores")
```

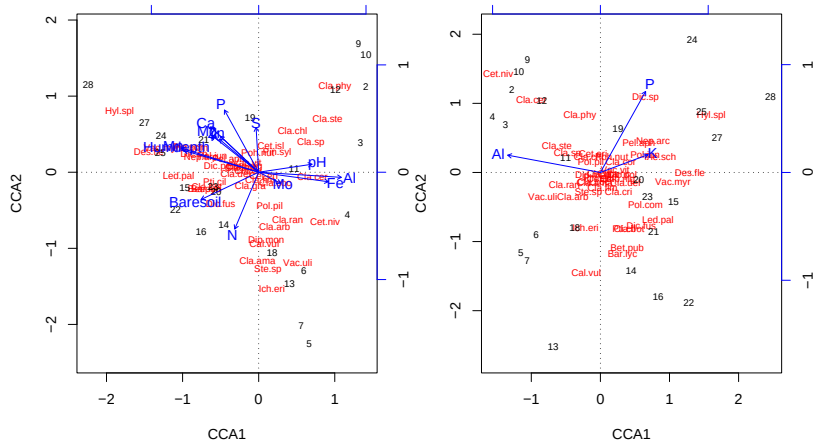
The graphs are (almost) identical, but this may be difficult to see, because orientations differ, We cannot simply label the sites because they were shuffled, but the environmental variables were kept unchanged and *they* determine the configuration. We can analyses the data further with Procrustes analysis (slide 117):

```
> procrustes(scores(orig, dis="lc"), scores(shuff, dis="lc"))

Call:
procrustes(X = scores(orig, dis = "lc"), Y = scores(shuff, dis = "lc"))

Procrustes sum of squares:
0.007747
```

Function `procrustes` would extract WA scores as default, and therefore we must use the `scores` function explicitly. The difference is very small, and the configurations are practically identical. In `rda` they would be identical within numerical precision, but the weights change when shuffling the species data and this influences in the weighted regression (slide 130).

Slide 139

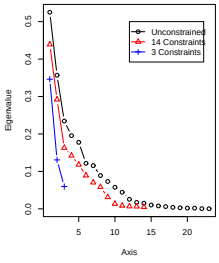
Number of constraints and the plot



Slide 140

Number of constraints and curvature

- Curvature cured because forced to linear constraints.
- High number of constraints = no constraint.
- Absolute limit: Number of constraints = $\min(S, N) - 1$, but release from the constraints can begin much earlier.
- Reduce environmental variables so that only the important remain: Heuristic value better than statistics.
- Reduces multicollinearity as well.



Note on slide 140. Ter Braak demonstrated that constraining removes the curvature in CA (slide 100) without need of detrending or other 'dirty tricks' [44, 88]. Moreover, he said clearly that this happens only with a relatively low number of constraints, and the curvature may re-appear when increasing number of environmental variables. The absolute upper limit is $\min(S, N) - 1$ with certainly no constraints. However, the release from the constraints in the first axes can begin much earlier. In fact, the arc effect or the curvature can re-appear with two unfortunately chosen continuous constraints or with a single three-class factor (see slide 143).

Slide 141

DECORANA in Disguise

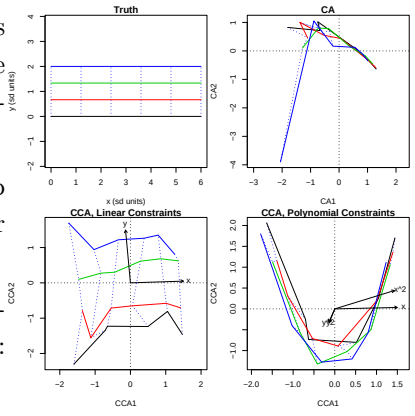
Constrained Correspondence Analysis replaced DECORANA as the canonical method — and indeed, it is DECORANA in disguise

- **Detrending:** Based on fitted values from linear regression
- **Rescaling:** Linear combinations of environmental variables – scaled similarly
- **Downweighting:** Rare species fit poorly

Slide 142

Polynomial Constraints: A Bad Idea

- Unconstrained CA produces curves, because species have non-linear responses to gradients
- Constrained CA straightens up curves, because it forces linear species responses
- Polynomial constraints produce quadratic fitted values: Ordination will be quadratic.

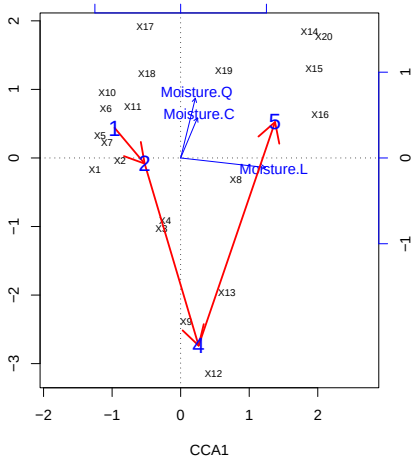


Note on slide 144. Functions `cca` and `rda` in vegan find all axes, constrained and unconstrained instead of only 4 or 3 first, and the unconstrained component is always available. The model `formula`e can contain a special term `Condition` for conditioning variables to

Slide 143

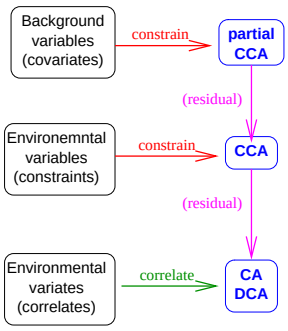
Constrained horseshoe

- Curve is removed in CCA because the solution is forced to linear constraints
- If contraints have a quadratic relation to each other, a curve may re-appear
- Polynomial constraints and interactions are generally a bad idea



Slide 144

Levels of environmental intervention



- **Partial CCA** removes the effect of background variables before proper (C)CA: 'random' or 'nuisance' variables.
- Residual ordinations may be analysed at all level: Partitioning of variation.
- Constraints are linear: If levels of environmental variables are not orthogonal, this may result in negative 'components of variation'.
- Information of lower levels mixed with upper.

be 'partialled out' before the analysis. So the following studies the effect of Calcium and Magnesium in the soil 'partialling out' the effects of pH:

```
> cca(varespec ~ Ca + Mg + Condition(pH), varechem)

Call:
cca(formula = varespec ~ Ca + Mg + Condition(pH), data = varechem)

              Inertia Rank
Total          2.0832
Conditional    0.1458    1
Constrained    0.2478    2
Unconstrained  1.6895   20
Inertia is mean square contingency coefficient

Eigenvalues for constrained axes:
   CCA1    CCA2
0.18330 0.06455

Eigenvalues for unconstrained axes:
    CA1     CA2     CA3     CA4     CA5     CA6     CA7     CA8
0.38339 0.27366 0.20890 0.17580 0.15731 0.11604 0.10459 0.07437
(Showed only 8 of all 20 unconstrained eigenvalues)
```

There is no direct 'partitioning of variation' in `vegan`, but all the information is readily available.

There is an abundant literature about 'partitioning of variation' [15, 70, etc.]. There are several problems and pitfalls in this partitioning, and I should write some slides about this. Not today, but you just wait and see...

Partial analysis can be used to compare two constrained variables, and to estimate the 'significance' (slide 145) of their difference in the same way as in ANOVA. The effect of adding term `N` to a model already having terms `P` and `K` (the symbols refer to major nutrients in soil) can be analysed in this way:

```
> anova(cca(varespec ~ N + Condition(P + K), varechem))
Permutation test for CCA under reduced model

Model: cca(formula = varespec ~ N + Condition(P + K), data = varechem)
         Df  Chisq      F N.Perm Pr(>F)
Model     1 0.1141 1.3943    100   0.21
Residual 20 1.6368
```

Note on slide 145. CANOCO makes permutation tests either for the first eigenvalue or for the sum of all eigenvalues; only the latter is available in `vegan`. In principle, tests could be performed like in the graph in the slide for any axis, but I know of no canned application (the graph was done in R, so this is doable in R). The meaning of such a test is not quite clear either, neither is the meaning of testing only for the first eigenvalue.

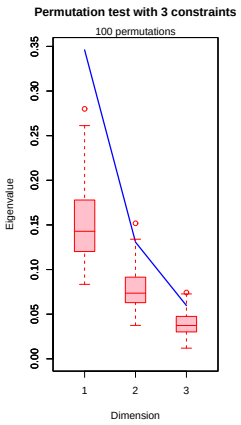
The permutation tests are made in `vegan` using function `anova.cca` that can handle either `rda` or `cca`:

```
> vare.cca <- cca(varespec ~ Al + P + K, varechem)
> anova(vare.cca)
Permutation test for CCA under reduced model

Model: cca(formula = varespec ~ Al + P + K, data = varechem)
         Df  Chisq      F N.Perm Pr(>F)
Model     3 0.6441 2.9840    100 < 0.01 ***
Residual 20 1.4391
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Slide 145

Significance of constraints

- CCA maximizes eigenvalue with constraints.
- Permutation tests can be used to assess significance:
  - Permute lines of environmental data.
  - Repeat CCA with permuted data.
  - If observed λ higher than (most) permutations, regarded as significant.
- Many constraints = much opportunity for optimizing: Significance usually lower.



The function finds the number of permutation cycles with a simple heuristics: It runs a specified number of permutation cycles (default 100), and if the observed significance is probably different from the target level (0.05), it stops, but otherwise it continues with the same number of steps. In this way the permutations stop early in clear cases, but more work is done in dubious cases.

Slide 146

Permutation statistic

- Without constraints, sum of all eigenvalues is a natural choice
- Testing of first eigenvalue has an unclear meaning
- In partial models, use "pseudo-F":

$$F_{p,q} = \left(\sum_i^p \lambda_i^{(c)}/p\right) / \left(\sum_i^q \lambda_i^{(r)}/q\right)$$

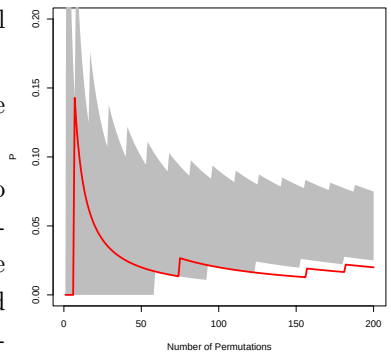
  with constrained (c) and residual (r) eigenvalues λ and respective number of axes p and q.
- Not at all distributed like real F, but used in permutation tests

Note on slide 150. The three examples were all built using a preliminary `step.cca` function with `cca`. The first panel stepped forward and the `scope` included all main effects in `varechem`. The second stepped forward, but its `scope` allowed first order interactions. The last started with all variables of `varechem` and used backward elimination. In the first case, the variables entered in the order `Al`, `P`, `K`. When the first order interactions were allowed, the fourth term was `P:K` interaction and this made `Baresoil` to appear as important – so these two models differ in main effects as well. The final formula can be written as

Slide 147

Number of permutations

- Too few permutations: Cannot detect "significant" response when it is close to a critical limit
- Too many permutations waste time
- Sequential testing: Permute so many times that assessed significance is "certainly" outside the grey zone where it could be either "significant" or "non-significant"



Slide 148

What is permuted?

- No conditioning variables: Community data or constraints can be permuted
- In partial models ith conditioning variables:
  - Community data cannot be permuted, because it is dependent on conditions
  - Constraints cannot be permuted, because they correlate with conditions
- Residuals are exchangeable *if they are independent and identically distributed...*
- **Reduced model** permutes residuals after conditions, **Full model** residuals after conditions and constraints

Slide 149

Selecting constraining variables

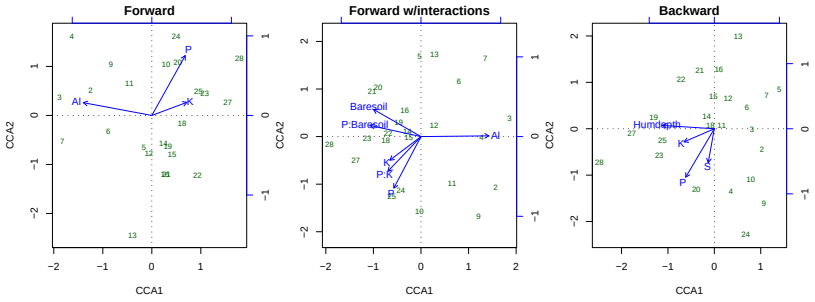
- Small number of variables means stricter constraints, reduced curvature, improve interpretation, increases signficance: Try to end up with one or two constraints for each independent factor.
- Significance tests and visual inspection *help* in selecting environmental variables.
- Automated selection dangerous: Small changes in data set can change the whole selection history, and omission of a variable does not mean it is unimportant.
- Final selection must be made with heuristic criteria.

The purpose of computation is insight, not numbers

Slide 150

Automatic stepping is dangerous

Automatic model selection may give different results depending on stepping direction, scope or small changes in the data set



`cca(varespec ~ Al + P*(Baresoil + K), data=varechem)` in `vegan`. Aluminium (`Al`) was the first term to enter in forward stepping, but it was the second to eliminated (after `Fe`) in backward elimination, and so these two 'best' main effect models differ strongly. The graphs use LC scores.

The `step.cca` may appear in the future versions of `vegan`, but I do not guarantee this, because I think that automatic stepping is a bad idea. However, all the tools needed are already there, but the model building must be made by hand.

Slide 151

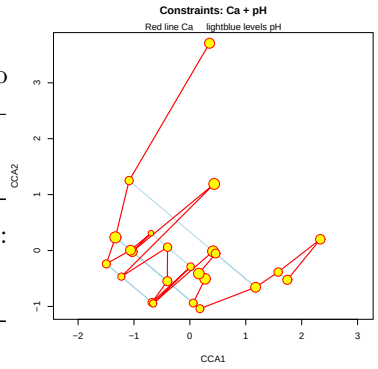
Components of Variation

- Take a partial model CCA($Y \sim X|Z$)
- The explained Inertia can be decomposed into two components:
  1. Explained by X in a simple model CCA($Y \sim X$)
  2. The residual effect of X after removing the variation caused by the conditioning variable Z
- After conditioning by Z, the eigenvalue of X decreases by the amount of shared component of variation

Slide 152

Negative Components of Variation

- CCA($Y \sim X|Z$) is equal to CCA(Res(CCA($Y \sim Z$)) $\sim X + Z$)
- If variables are better predictors together than in isolation: $\lambda_{X+Z} > \lambda_X + \lambda_Z$
- Constraints allow the reappearance of the curve



$\lambda_{\text{Ca}} = 0.157$

$\lambda_{\text{Ca|pH}} = 0.183$

5 Gradient Model and Ordination

A good method should give us an image of reality. For that we must assume that there is a real pattern. So we need a model of vegetation. The most commonly used model is the gradient model (slide 18). With this model, we think that a good model is able to recover the gradient structure from the community composition. For comparing methods, we need to first know the real structure. This means that we cannot use observed data sets, because there we do not know the reality, but we infer it from the very same data we should use for comparing methods. It was common in olden times to do such method comparisons with observed data, but they were all circular and flawed. So wee need to simulate data with known structure, and see how different methods can find that structure. Only after that we can assume that if similar patterns are found in observed data, similar gradient structure could be latent in them.

Slide 153

Comparing methods

- Ordination methods cannot be compared with real data sets: The truth is unknown.
  - The correct structure is inferred from the data, and the comparison biased towards the pet result.
- Comparison needs external criteria (environmental variables?).
- Simulated community pattern:
  - Assume an interpretable gradient pattern, and see if the method can find this pattern: Reliable only if it finds the known pattern.
  - Robustness is the ability to work even when the assumptions are violated.

Note on slide 153. Minchin [61] set the modern standards of simulating community patterns. He introduced the concept of robustness to comparison (although this concept was misapplied since then by other authors). In the context of vegetation simulation, robustness means that a method is able to find the gradient structure with any kind of response models. Further, simulations must be replicated with similar parametrization to see that they can be relied upon. It is important to use noisy data in all simulations.

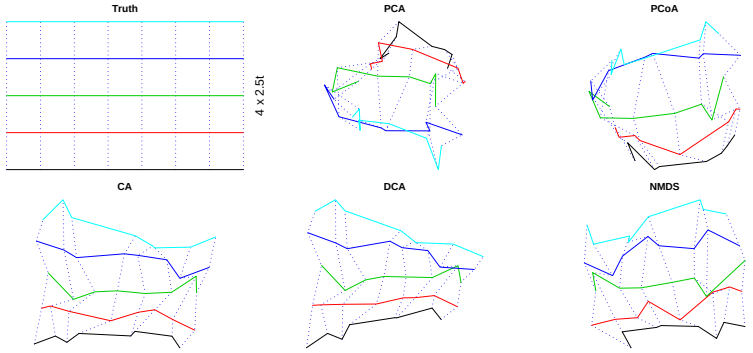
Michin wrote programme Compas for community pattern simulation. It is still the standard tool of simulation, but may be difficult to find. Dave Roberts wrote programme CoenoFlex for a more flexible community pattern simulation. It is available through his web page http://labdsv.nr.usu.edu/.

Note on slide 154. These community pattern simulations are very simplistic, and made solely for producing nice graphs for these lectures. I used only Gaussian response functions, binary (presence/absence) data and Binomial error. The gradient lengths are defined using the Gaussian width parameter, tolerance or 'sd' (slide 23).

Note on slide 155. I call this 'folklore', because this is not based on research: Minchin's [60] showed that with short gradients PCA was about as good as CA, but in all cases it was weaker. In statistic, such a method is called 'uniformly weaker.' Further, the length of the gradients cannot be reliably inferred from ordination results, and there is no way of deciding

Slide 154

Community pattern simulation

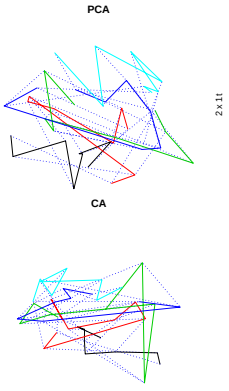


Gaussian response model, presence/absence, binomial(1) error.

Slide 155

Short gradients: Is there a niche for PCA?

- Folklore: PCA with short gradients ($\leq 2t$).
- Not based on research, but simulation finds PCA uniformly worse than CA: At the best case about as good as CA.
- There should be no species optimum within gradient: Shortness alone not sufficient.
- PCA best used for really linear cases (environment) or for reduction of variables into principal components (but see FA).
- Noise dominates over signal in homogeneous data.

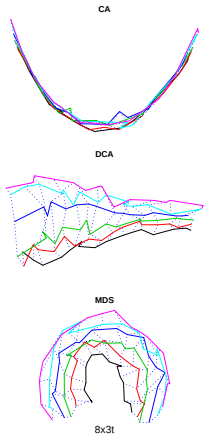


when we should switch away from linear methods. However, this advice is repeated in most textbooks on ordination.

Slide 156

Long gradients: DCA or NMDS

- Curvature with long gradients: Need either DCA or NMDS.
- NMDS is a test winner: More robust than DCA.
- DCA more popular.
- DCA may produce new artefacts, since it twists the space.



Slide 157

Handling curves

1. **Accept and acknowledge**: Clear only with a single dominant gradient.
2. **Degree of absence**: If curve caused by 'noughty noughts', estimate how much the species is absent.
3. **Extended dissimilarity**: If all distant points have nothing in common, estimate dissimilarity through stepstone points.
4. **Detrend**: Distort the space so that configuration disappears.
5. **Monotone regression**: Do not require linear relation with ordination and dissimilarity.
6. **Constrain**: Use linear constraints to force straigthening.

Note on slide 157.

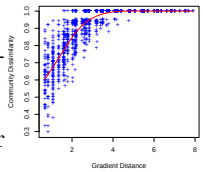
1. 'Accept and acknowledge' is successfull only with one-dimensional data when it is possible to see that there is a curve: Feoli, Orlóci, Mike Dale...
2. 'Degree of absence' was introduced by Swan in 1971. It was never very popular, despite being an ingenious concept. However, Sambuk in Russia (then U.S.S.R.) used that extensively.
3. 'Extended dissimilarity' used to be a historic concept, used only in one rarerely [16, 102], but it resurrected recently [19], and is now regarded as 'hot'. Dave Roberts has implemented this in his `dsvdis` function in `labdsv`.

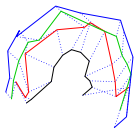
4. 'Detrend' used to be the standard answer, and there are still many devoters (slide 99).
5. 'Monotone regression' is the core of NMDS (slide 111). This is still a popular alternative, and probably the best uncostrained ordination method.
6. 'Constrain' is now the standard answer, since constrained or 'canonical' correspondence analysis *seems* to remove the curvature without any dirty tricks, and with some extra benefits. More from slide 129 onwards.

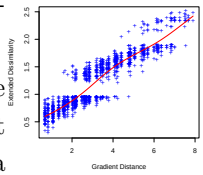
Slide 158

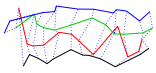
Extended dissimilarities and step-across

- How different are sites that have nothing in common?
- Use step-across points to estimate their distance
- Flexible shortest path or their approximations, extended dissimilarities
- Extended dissimilarity: use only one-site steps, do not update dissimilarities below a threshold



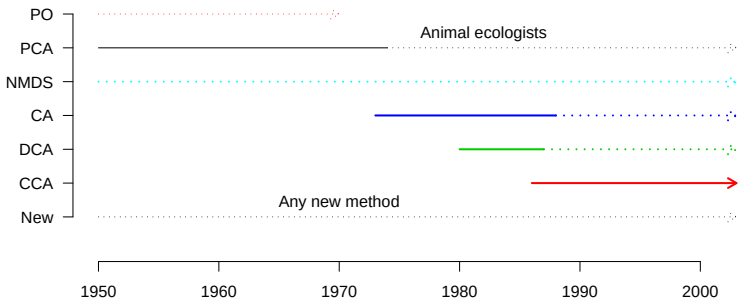






Slide 159

Analysed using modern methods...



Note on slide 159. One of the methods in the graph was not mentioned in these lectures: PO or Polar Ordination. It is a primitive variant of MDS where two extreme points are selected and all other are arranged between these 'polar points' according to their dissimilarities. For a second axis, two points close to each other on the first axis, but dissimilar

according to indices, are selected as the second 'polar points' and the ordering is repeated. The method is of historic interest only, and it was used before computers were available. However, some people still propagate for the method, and have suggested new developments which usually mean that the method loses its only attraction: simplicity [59].

Fancy, modern methods never end. Today they may be Self Organizing Maps, Principal Curves [20], Semi-strong Hybrid Scaling (Belbin) Co-Inertia Analysis [21]...

6 Classification

This chapter is badly in need of upgrading

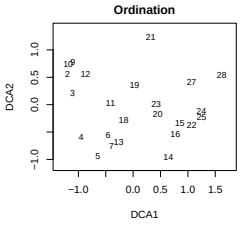
Classification and ordination were seen as antagonistic strategies in the past. Ordination presented modern, dynamic approach, whereas classification was doomed and reactionary. Nowadays they are alternative methods of capturing multivariate structure, and often used in parallel.

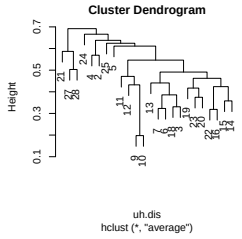
R has alternative libaries for classification analysis. Library `mva` has the basic hierarchic methods (function `hclust` with support functions). Fancier methods [46] are collected in library `cluster`. Dave Roberts's `labdsv` library and Jean Thioulouse's `ade4` have a better support for community ecological methods. These lecture notes are (too) short on classification, and the methods in `hclust` are mainly sufficient.

Slide 160

Classification

- Nobody should **want** to make clustering, but they are desperate with multivariate data.
- Reduce data into a few classes and describe these instead of original observations.






```
                  11111111122222222
         23456790123456890123456 78
Cal.vul ++5.1..14185352.+.5.+...
Emp.nig 652+665737461676675746 74
Led.pal ..........+......54...42
Vac.myr ......1...1+24.4.76...67
Vac.vit 654566777766757678787776
Vac.uli .4+..6........4.2.4..+31
Dic.sp  .........1+.23..31+.87.1
Dic.fus +2113221114778414185.752
Dic.pol ...+.+111...+1.++5.+4..1
Hyl.spl ................4....+66
Ple.sch 3+4+++537666875774878899
Pol.jun +.+2+1111.+11++4+1.416.+
Pti.cil ++..+.1.3.1.+++1.6+4+1++
Cla.arb 267788236587768663665641
Cla.ran 887989568784678575566463
Cla.ste 99837199897+1+58+71++.4.
Cla.unc 113152114338531151436 3++
Cla.cor +1+1111111+21+11+++4122+
Cla.cri ++++1+111123311++114++++
Cet.niv .+61+131......++...+....
Ste.sp  .136+3.13.+23+21...4++.+
Cla.def +++11+11.124331232+42+++
```

Slide 161

Classification of classification

Formal	$\Longleftrightarrow$	Informal
Hierarchic	$\Longleftrightarrow$	Non-hierarchic
Quantitative	$\Longleftrightarrow$	Qualitative
Divisive	$\Longleftrightarrow$	Agglomerative
Polythetic	$\Longleftrightarrow$	Monothetic

- **Cluster analysis:** Formal, hierarchic, quantitative (usually), agglomerative, polythetic.
- **TWINSPAN:** Formal, hierarchic, semi-quantitative, divisive, polythetic.

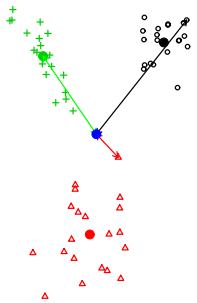
6.1 Cluster analysis

Cluster analysis is the old and traditional classification method in statistics, and even in community ecology. Clustering needs dissimilarity matrix as an input, and it builds a hierarchic tree where all observations are finally united at the root (ecologists usually draw this tree inverted or felled, so that the root is at the or on the left).

Slide 162

Cluster Analysis

- Agglomerative: Combine two most similar observations, and continue until every point is in the tree.
- Various criteria for similarity between clusters:
  1. Single linkage or distance to the nearest neighbour.
  2. Complete linkage or distance to the furthest neighbour.
  3. Average linkage or distance to the class centroid.



Note on slide 162. All these basic tree building methods can be used in the `hclust` function (library `mva`), but library `cluster` provides same alternatives — and many more. The `hclust` function needs dissimilarity index as the input. Functions `vegdist` (`vegan`) or `dsvdis` (`labdsv`) provide dissimilarity indices for community ecologists, in addition to basic indices in `dist` (`mva`) or in `daisy` (in library `cluster` which in general uses strange names). All these functions return similar objects which can be substituted for each other in analyses.

In clustering we must study the formal properties of the indices [54], and the properties required depend on tree building strategy (see slide 85). Average linkage methods estimate the dissimilarities among centroids, and indices must be such that those centroids can be found with simple calculations. Euclidean distance (or dissimilarity) or, in some methods, squared Euclidean distance usually satisfy the criteria in the centroid methods. On the other hand, single linkage methods may be more permissive for indices. If a 'non-metric' or other unsuitable index is used, it may appear that clusters at higher fusion level falsely appear as more uniform than smaller groups below them, and this may result in inversed branches in trees.

Note on slide 163. All examples were made using the `hclust` function (library `mva`), but library `cluster` provides the same alternatives (and some more) in an strangely named function `agnes`. The methods have a `plot` method for displaying dendrograms:

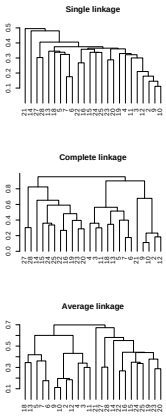
```
> vare.dist <- vegdist(wisconsin(varespec), "bray")
> vare.hc <- hclust(vare.dist, "complete")
> plot(vare.hc)
```

Note on slide 164. Hierarchic clustering produces a tree (dendrogram) which can be cut at any level. In fact, there are one less of joints than there were observations. Calling

Slide 163

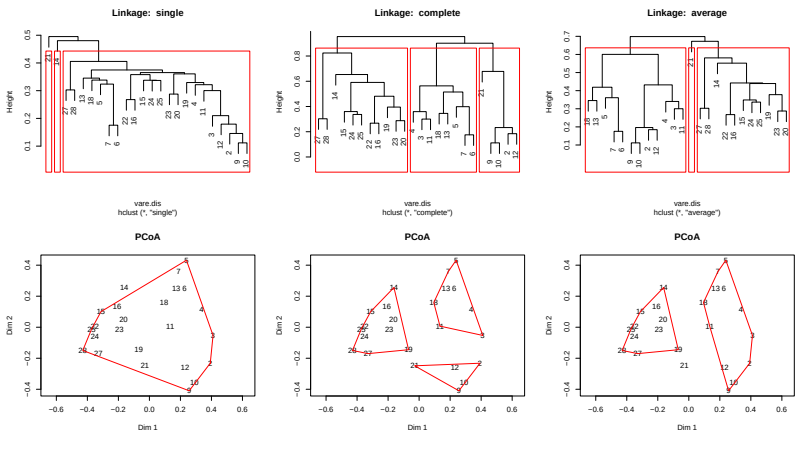
Clustering strategies

- **Single linkage or nearest neighbour**
  - Finds the *minimum spanning tree*: Shortest tree that connects all points.
  - Finds discontinuities.
  - Chaining: Groups of inequal size.
- **Complete linkage or furthest neighbour**
  - Compact clusters of ±equal size.
  - Makes compact groups even when none exist.
- **Average linkage methods** (e.g. UPGMA)
  - Between single and average linkage.
  - UPGMA minimizes *cophenetic correlation*.



Slide 164

Clustering and space



them classification may be somewhat misleading, because they do not give one classification, but several alternative classifications, with extreme cases of all observations in one cluster or each observation in its private cluster. Therefore the user commonly wants to cut the classification at a certain level, and pretend that this was *the* classification produced by the method.

R function `cutree` (with only one '`t`', library `mva`) cuts the classification to give a desired number of groups. Function `rect.hclust` can be used to display the grouping by drawing boxes in the tree separating clusters at a given number groups.

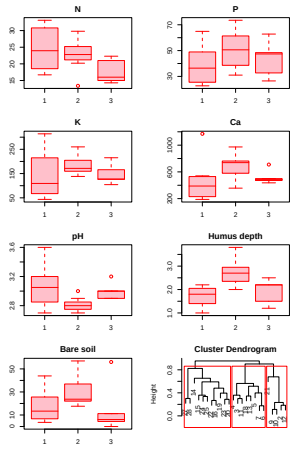
```
> cutree(vare.hc, 3)
18 15 24 27 23 19 22 16 28 13 14 20 25  7  5  6  3  4  2  9 12 10 11 21
 1  2  2  3  2  2  2  2  3  2  2  2  2  1  1  1  1  1  1  2  2  2  2  3
> rect.hclust(vare.hc, 3)
```

The ordination diagram depicts the groups using *convex hull*. A convex hull encloses all points so that no line between two points can be drawn outside the convex hull. In R, a convex hull can be found using the function `chull`.

Slide 165

Interpreting clusters

- Predictive clusters are different in their environment.
- Community classification for environmental indication.
- Clustering may detect local peculiarities, whereas (most) ordination methods show the global gradient pattern.



Note on slide 165. A basic boxplot can be drawn using:

```
> boxplot(Baresoil ~ cutree(vare.hc, 3))
```

The `cutree` classification can be used as a classifier (factor) in more formal tests of significance as well.

6.2 Other classification methods

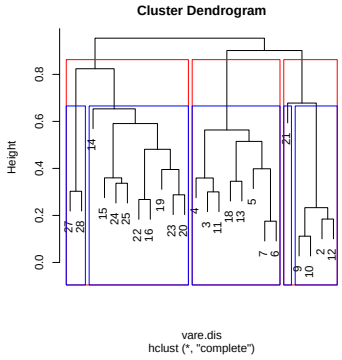
Library `cluster` has some alternatives to hierarchic clustering. Some of these try to find directly a certain number of clusters. Some people regard that as a good feature, because the scientist wants a classification at one level, and must arbitrarily cut a classificatin at some level. Others regard that as a 'subjective element' which makes these methods dubious, because the scientist must make a decision instead of blindly relying on 'objective' methods.

Note on slide 167. There are two alternatives of K–means clustering: Library `mva` has the function `kmeans`, and library `cluster` has strangely named function `pam`. The latter

Slide 166

Number of clusters

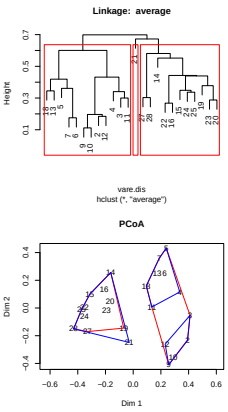
- As many fusion levels as there are observations: Hierarchic clustering can be cut at any level.
- The scientist usually want to use classes: One level.
- Various optimality criteria doomed to fail: If they are good, they can be made clustering criteria, and then they are just an alternative clustering.



Slide 167

Optimizing classification: K–means clustering

- Agglomerative clustering has a burden of history: Once formed classes cannot be broken although that would be sensible at the chosen level.
- K–means clustering: Iterative procedure for non-hierarchic classification.
- If started with chosen hierarchic clustering, will optimize.
- Best suited with centroid linking, since thinks in that way.



uses the medoid method which should be more robust than the basic K–means. Function `kmeans` was used for this slide.

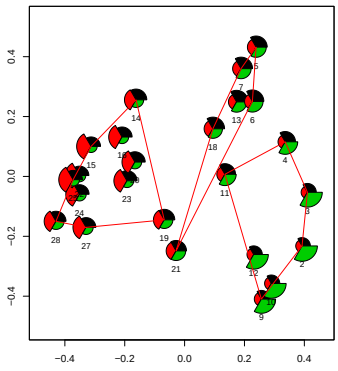
An old, useful programme in vegetation science was TabOrd [**?**] which may be difficult to find nowadays. It took a clustering at a given number of clusters and tried to improve that classification by moving points among groups, and possibly fusing groups. Something similar can be achieved with these methods.

I should add something about silhouette plots...

Slide 168

Fuzzy clustering

- Each observation is given a probability profile of class membership.
- Corresponding crisp classification: Class of highest membership probability.
- Non-hierarchic, flat classification.
- Iterative procedure.



Note on slide 168. Community ecologists have criticized clustering because there are no discrete types in Nature. On the other hand, they have confessed that classes might be more practical in the field than coordinates of ordination space. Fuzzy clustering seems to combine the good sides of classification and ordinations: Gives you classes to use in the field, but doesn ot pretend them to be natural entities.

Fuzzy clustering can be made with strangely named function `fanny` (library `cluster`). Function `fanny` can take either a data matrix directly or it can use some types of dissimilarities. The number of classes must be defined in the call:

```
> vare.fuzz <- fanny(varespec, k=3)
```

Some versions of `fanny` have been unreliable, and all versions have not worked with all options in all data sets. It is recommended to upgrade to the latest version of `cluster` in all cases.

Function `fanny` has its own plotting methods, but the graph in the slide is grafted by hand in the following lines:

```
> vare.pcoa <- cmdscale(vare.dist)
> plot(scores(vare.pcoa), asp=1, type="n")
> stars(vare.fuzz$membership, location=scores(vare.pcoa), draw.segments=T,
+ add=T,  scale=F, len=0.05)
```

Note on slide 169. Twinspan is poor and unreliable method that should not be used. It uses a kludge of pseudospecies, and has many other quirks so that its analyses may impossible to repeat. The results are unstable and small changes in the data or in the parametrization can cause great changes in the results. Twinspan is not available in R,

Slide 169

TWINSPAN: Two-Way Indicator Species Analysis

- TWINSPAN is not a method but a program: A bag of tricks.
- Gradient chopping for CA: Ideal if this is the criterion.
- Uses binary data: Trick is to divide each species into a series of 'pseudospecies' by abundance cuts.

Algorithm

1. Get a CA axis on pseudospecies data.
2. Select pseudospecies at the ends of the axis as indicators.
3. Repeat ordination with these pseudospecies: Polarizes the axis.
4. Chop data in two parts in the middle of the axis.
5. Repeat steps for both parts.

but you must get a stand-alone program which is freely distributed and readily available, for instance from my web pages as Windows and Linux binaries and in the source code.

There are better methods: Use them.

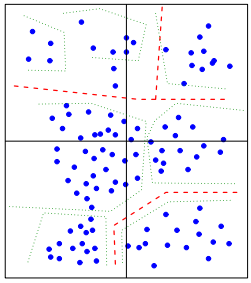
6.3 Comparing classification methods

It was possible to compare ordination methods: There is an underlying gradient model of vegetation, and a good ordination method can find those gradients (slide 153). However, there is no such a natural model for classification — at least nobody has proposed a classificatory model of vegetation. In lack of image of Nature, we cannot know what kind of image should a method draw. People have proposed several criteria of good classes (slide 170), but these may be contradictory, and we do not know whether Nature should follow these rules.

Slide 170

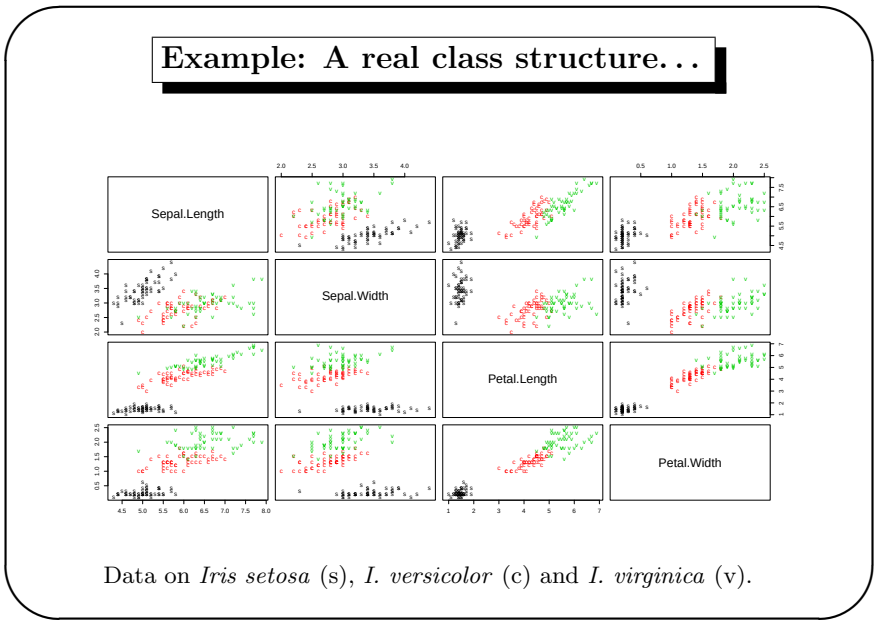
Criteria for good classes

1. Divide environment into equal parts.
2. Compact clusters.
3. Groups of equal size.
4. Discontinuous groups.



These criteria often in conflict, and cannot be satisfied simultaneously.

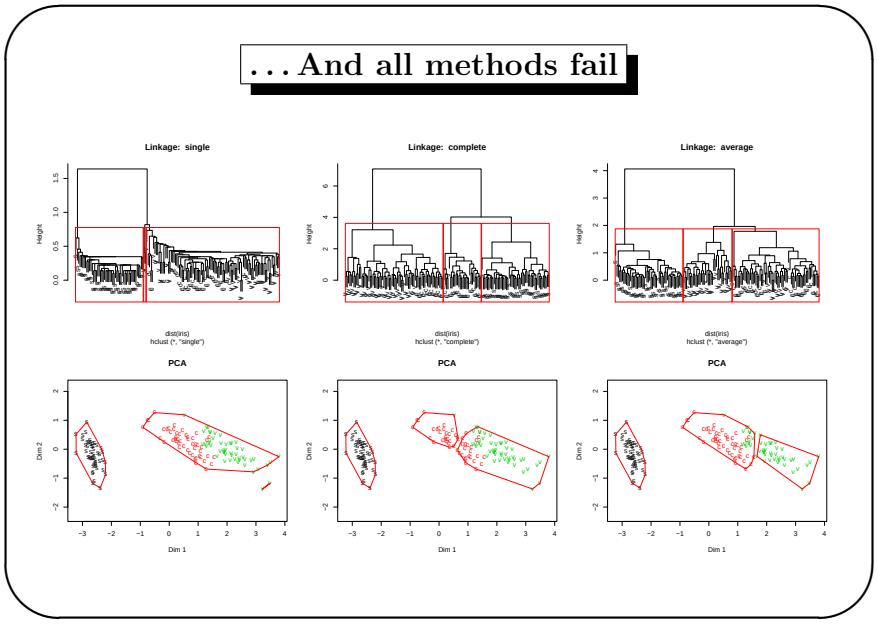
Slide 171



Note on slide 171. The example uses Anderson's (or Fisher's?) famous *Iris* data, which is available in the base R (data `iris`), so the graph was drawn as:

```
> data(iris)
> spec <- c("s","c","v")[codes(iris$Species)]
> plot(iris[,1:4], pch=spec, col=codes(iris$Species))
```

Slide 172



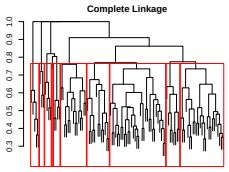
Note on slide 174. I admit, the end is abrupt: I was getting tired and busy to finish these slide before my lectures. Classification deserves a better treatment.

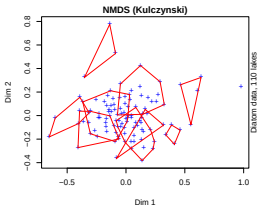
The triangle plot was drawn using R code I found floating in the hyperspace (search R News Archives), but `ade4` library has a much better function for the task.

Slide 173

Classification and ordination

- Formerly classification and 'continuum' theoretical ordination were seen as opposites: Only two alternative ways of simplification of multivariate data.
- If classes distinct in ordination, results (or methods!) are consistent.
- Inconsistent results:
  - Either or both results bad.
  - Different criteria.
  - Too few dimensions in ordination.

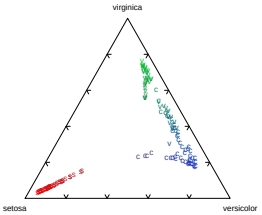




Slide 174

The choice of clustering method

- Some opt for single linkage: Finds distinct clusters, but prone to chaining and sensitive to sampling pattern.
- Most opt for average linkage methods: Chops environment more evenly.
- All dependent on dissimilarity measure: Should be ecologically meaningful.
- Small changes in data can cause huge visual change in clustering: Classification may be optimized for the chosen level.
- TWINSPAN too unstable and tricky: Better avoided.



Fuzzy clustering may fail as well, but at least shows the uncertainty.

A Appendix

A.1 Data Import

R distribution usually includes a more complete documentation "*R Data Import/Export*". This chapter gives only brief hints for the starters.

Slide 175

Data Import to R

- Read "*R Data Import/Export*"
- Import from SPSS, SAS *etc*: Standard library `foreign`
- CEP data: Function `read.cep` in library `vegan`
- Spreadsheet:
  1. Edit for R
  2. Save as *comma separated values* (CSV)
  3. Read with `read.csv` (decimal points, separators "`,`") or `read.csv2` (decimal commas, separators "`;`")

Note on slide 175. Package `foreign` is a recommended package that should be included in all R installations. The proprietary formats like SPSS may change without warning, and all data sets may not be readable.

Function `read.cep` in `vegan` should be able to handle the same file types as CANOCO: condensed, fixed width rectangular formats and `FREE` formats. However, these are only a subset of the original CEP formats [28], but as the other types are not accepted by CANOCO, they are rarely used nowadays. Function `read.cep` used to crash R in WINDOWS, but it seems to work in modern installations. However, I do not know if it works in old R or in old versions of WINDOWS: Save your data and session (`save.image()`) before using the function.

Note on slide 176. The first column should be species names, and the first row should be the site names. The upper corner (cell `A1`) is left empty.

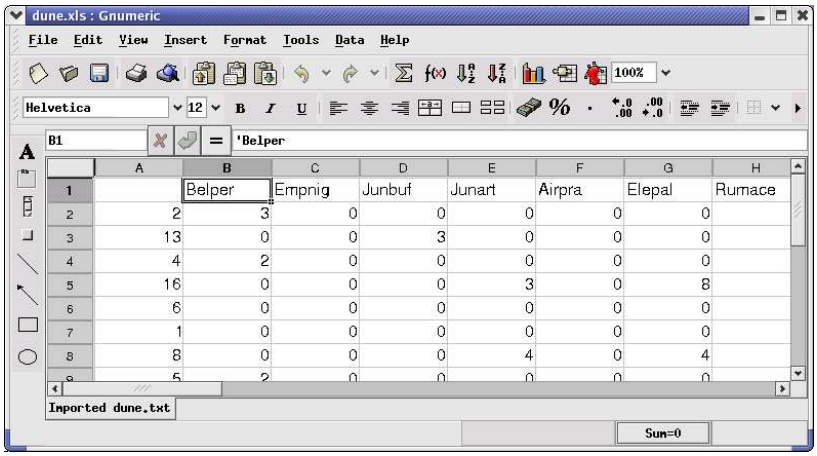
Note on slide 177. When the spreadsheet of slide 176 is saved as a CSV file, there may be an extra comma before the first name (and after the empty cell). This comma should be removed in a text editor (such as NOTEPAD) before reading in the data.

Note on slide 178. One of the most common problems in data import is that R regards some variables as `factor`s although they were supposed to be all numeric. This happens if R finds any entry that is not recognized as a number. Commonly this is an empty cell. See functions `factor` and `as.factor` for changing the variables into factors. In particular, you may give the order of `levels` or define a factor as `ordered`.

Note on slide 179. Underscore is not allowed in names because in the original S language it was used as an assignment sign instead of "`<-`" or current "`=`" (which may not work in all situations). The use of "`_`" is currently deprecated in R but it still illegal in names. However,

Slide 176

Preparing a spreadsheet



Slide 177

Comma separated values

```
Belper,Empnig,Junbuf,Junart,Airpra,Elepal,Rumace
2,3,0,0,0,0,0
13,0,0,3,0,0,0
4,2,0,0,0,0,0
16,0,0,0,3,0,8
6,0,0,0,0,0,6
1,0,0,0,0,0,0
8,0,0,0,4,0,4
5,2,0,0,0,0,5
```

NB you may have to remove a comma (`,`) before the first species name

Slide 178

Community data and Environmental data

- Best to keep in two separate files
- The order of sites must be identical in both data sets
- Factor variables best coded with informative names
- Do **not** use uninformative numbers or "dummy" variables
- Species data must be numeric: use zero (`0`) for missing species, since blanks default to missing values (`NA`).

Slide 179

Names

- It is best to name both variables (columns) and observations (rows), or R will make up uninformative names for both
- Names should be informative but short to avoid congestion
- Convention for species names: $4 + 4$ from generic and specific names (`Asiootus`, `Tytoalba`, `Bubobubo`)
- Avoid gaps (blanks), mathematical symbols (`+-*:*^`), accented characters (`øäñ`) and underscore (`_`) — but "`.`" is OK
- R will change non-conforming names (`make.names`)

if it, or any other unsuitable character, occurs in the input data, R changes that into an acceptable character, probably to "." using function `make.names`.

Names can be used as indices, and two data sets can be merged by their names. Therefore it makes sense to use consistent names, or standard database names for species.

References

[1] D. A. Anderson. Some models for overdispersed binomial data. *Australian Journal of Statistics*, 30:125–148, 1988.

[2] M. P. Austin. On non-linear species response models in ordination. *Vegetatio*, 33:33–41, 1976.

[3] M. P. Austin. Searching for a model for use in vegetation analysis. *Vegetatio*, 42:11–21, 1980.

[4] M. P. Austin. Models for the analysis of species' response to environmental gradients. *Vegetatio*, 69:35–45, 1987.

[5] M. P. Austin. On silent clash of paradigms: some inconsistencies in community ecology. *Oikos*, 86:170–178, 1999.

[6] M. P. Austin, R. B. Cunningham, and P. M. Fleming. New approach to direct gradient analysis using environmental scalars and statistical curve-fitting. *Vegetatio*, 55:11–27, 1984.

[7] M. P. Austin and M. J. Gaywood. Current problems of environmental gradients and species response curves in relation to continuum theory. *Journal of Vegetation Science*, 5:473–482, 1994.

[8] M. P. Austin and J. A. Meyers. Current approaches to modelling the environmnetal niche of eucalypts: Implications for management of forest biodiversity. *Forest Ecology and Management*, 85:95–106, 1996.

[9] M. P. Austin and A. O. Nicholls. To fix or not to fix the species limits, that is the ecological question: Response to Jari Oksanen. *Journal of Vegetation Science*, 8:743–748, 1997.

[10] M. P. Austin, A. O. Nicholls, M. D. Doherty, and J. A. Meyers. Determining species response functions to an environmental gradient by means of a β-function. *Journal of Vegetation Science*, 5:215–228, 1994.

[11] M. P. Austin and T. M. Smith. A new model for the continuum concept. *Vegetatio*, 83:35–47, 1989.

[12] J. Barkman. *Phytosociology and ecology of cryptogamic epiphytes*. Van Gorcum, Assen, 1958.

[13] A. M. F. Bio, R. Alkemade, and A. Barendregt. Determining alternative models for vegetation response analysis: a non-parametric approach. *Journal of Vegetation Science*, 9:5–16, 1998.

[14] H. J. B. Birks, J. M. Line, S. Juggins, A. C. Stevenson, and C. J. F. ter Braak. Diatoms and pH reconstruction. *Philosophical Transactions of the Royal Society London B*, 327:263–278, 1990.

[15] D. Borcard, P. Legendre, and P. Drapeau. Partialling out the spatial component of ecological variation. *Ecology*, pages 1045–1055, 1992.

[16] G. E. Bradfield and N. C. Kenkel. Nonlinear ordination using flexible shortest path adjustment of ecological distances. *Ecology*, 68:750–753, 1987.

[17] N. E. Breslow. Extra-Poisson variation in log-linear models. *Applied Statistics*, 33:38–44, 1984.

[18] M. J. Crawley. *GLIM for ecologists*. Blackwell, Oxford, 1993.

[19] G. De'ath. Extended dissimilarity: method of robust estimation of ecological distances with high beta diversity. *Plant Ecology*, 144:191–199, 1999.

[20] G. De'ath. Principal Curves: a new technique for indirect and direct gradient analysis. *Ecology*, 80:2237–2253, 1999.

[21] S. Dolédec and D. Chessel. Co-inertia analysis: an alternative method for studying species-environment relationships. *Freshwater Biology*, 31:277–294, 1994.

[22] O. Eilertsen, R. H. Økland, T. Økland, and O. Pedersen. Data manipulation and gradient length estimation in dca ordination. *Journal of Vegetation Science*, 1:261–70, 1990.

[23] R. Ejrnæs. Can we trust gradients extracted by detrended correspondence analysis? *Journal of Vegetation Science*, 11:565–572, 2000.

[24] H. Ellenberg. *Vegetation Mitteleuropas mit den Alpen in ökologischer Sicht*. E. Ulmer, Stuttgart, 3 edition, 1983.

[25] O. Eriksson. The species-pool hypothesis and plant community diversity. *Oikos*, 68:371–374, 1993.

[26] D. P. Faith, P. R. Minchin, and L. Belbin. Compositional dissimilarity as a robust measure of ecological distance. *Vegetatio*, 69:57–68, 1987.

[27] R. A. Fisher, A. S. Corbet, and C. B. Williams. The relation between the number of species and the number of individuals in a random sample of an animal population. *Journal of Animal Ecology*, 12:42–58, 1943.

[28] H. G. Gauch, Jr. *Multivariate analysis in community ecology*. Cambridge University Press, Cambridge, 1982.

[29] H. G. Gauch, Jr. and R. H. Whittaker. Coenocline simulation. *Ecology*, 53:446–451, 1972.

[30] M. Greenacre and T. Hastie. The geometric interpretation of correspondence analysis. *Journal of the American Statistical Association*, 82:437–47, 1987.

[31] M. J. Greenacre. *Theory and applications of correspondence analysis*. Academic Press, London, 1984.

[32] T. J. Hastie and R. J. Tibshirani. *Generalized additive models*. Chapman & Hall, London, 1990.

[33] E. Heegaard. The outer border and central border for species environmental relationships estimated by non-paremtric generalized additive models. *Ecological Modelling*, 157:131–139, 2002.

[34] E. Heegaard and H. H. Hangelbroek. The distribution of *Ulota crispa* in relation to both dispersal- and habitat-related factors. *Lindbergia*, 24:65–74, 1999.

[35] R. Hilborn and M. Mangel. *The ecological detective: Confronting models with data*. Princeton UP, Princeton NJ, 1997.

[36] M. O. Hill. Diversity and evenness: a unifying notation and its consequences. *Ecology*, 54:427–432, 1973.

[37] M. O. Hill. Reciprocal averaging: an eigenvector method of ordination. *Journal of Ecology*, 61:237–249, 1973.

[38] M. O. Hill. DECORANA*: A* FORTRAN *program for detrended correspondence analysis and reciprocal averaging*. Cornell University, Ithaca, NY, 1979.

[39] M. O. Hill and H. G. Gauch, Jr. Detrended correspondense analysis: an improved ordination technique. *Vegetatio*, 42:47–58, 1980.

[40] Z. Hubálek. Coefficients of association and similarity, based on binary (presence–absence) data: an evaluation. *Biological Review*, 57:669–689, 1982.

[41] S. P. Hubbell. *The unified neutral theory of biodiversity and biogeography.* Princeton Univ. Press, 2001.

[42] J. Huisman, H. Olff, and L. F. M. Fresco. A hierarchical set of models for species response analysis. *Journal of Vegetation Science*, 4:37–46, 1993.

[43] S. H. Hurlbert. The nonconcept of species diversity: a critique and alternative parameters. *Ecology*, 52:577–586, 1971.

[44] R. H. Jongman, C. J. F. ter Braak, and O. F. R. van Tongeren. *Data analysis in community and landscape ecology.* Pudoc, Wageningen, 1987.

[45] G. Kantvilas and P. R. Minchin. Analysis of epiphytic lichen communities in tasmanian cool temperate rainforest. *Vegetatio*, 84:99–112, 1989.

[46] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introductin to Cluster Analysis.* J. Wiley, New York, 1996.

[47] R. Kay and S. Little. Transformations of the explanatory variables in the logistic regression model for binary data. *Biometrika*, 74:495–501, 1987.

[48] A. W. Kemp. Families of discrete distributions satisfying Taylor's power law. *Biometrics*, 43:693–699, 1987.

[49] M. Kent and P. Coker. *Vegetation description and analysis: A practical approach.* J.Wiley, Chichester, 1992.

[50] O. Klement. Prodromus der mitteleuropischen Flechtengesellschaften. *Feddes Repertorium Beihäfte*, 135:1–195, 1955.

[51] R. G. Knox. Effects of detrending and rescaling on correspondence analysis: solution stability and accuracy. *Vegetatio*, 83:129–136, 1989.

[52] J. E. Lawesson and J. Oksanen. Niche characteristics of Danish woody species as derived from coenoclines. *Journal of Vegetation Science*, 13:279–290, 2002.

[53] J. R. Leathwick. Are New Zealand's *Nothofagus* species in equilibrium with their environment? *Journal of Vegetation Science*, 9:719–732, 1998.

[54] P. Legendre and L. Legendre. *Numerical ecology.* Number 20 in Developments in Environmental Modelling. Elsevier, Amsterdam, 2nd edition, 1998.

[55] B. W. Lindgren. *Statistical theory.* Macmillan, New York, 1976.

[56] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate analysis.* Academic Press, London, 1979.

[57] P. McCullagh and J. A. Nelder. *Generalized linear models.* Chapman & Hall, London, 1989.

[58] B. McCune. Influence of noisy environmental data on canonical correspondence analysis. *Ecology*, 78:2617–23, 1997.

[59] B. McCune and E. W. Beals. History of the development of bray-curtis ordination. In J. S. Fralish et al., editor, *Fifty years of Wisconsin plant ecology*, pages 67–79. Wisconsin Academy of Sciences, Arts & Letters, Madison, 1993.

[60] P. R. Minchin. An evaluation of relative robustness of techniques for ecological ordinations. *Vegetatio*, 69:89–107, 1987.

[61] P. R. Minchin. Simulation of multidimensional community patterns: towards a comprehensive model. *Vegetatio*, 71:145–156, 1987.

[62] P. R. Minchin. Decoda: database for ecological community data. Anutech Pty Ltd., Canberra, 1988.

[63] P. R. Minchin. Montane vegetation of the Mt. Field massif, Tasmania: a test of some hypotheses about properties of community patterns. *Vegetatio*, 83:97–110, 1989.

[64] D. F. Moore. Asymptotic properties of moment estimators for overdispersed counts and proportions. *Biometrika*, 73:583–8, 1986.

[65] J. C. Nekola and P. S. White. The distance decay of similarity in biogeography and ecology. *Journal of Biogeography*, 26:867–878, 1999.

[66] S. Nishisato. *Analysis of categorical data: dual scaling and its applications*. University of Toronto Press, Toronto, 1980.

[67] R. Økland. Are ordination and constrained ordination alternative or complementary strategies in general ecological studies? *Journal of Vegetation Science*, 7:289–292, 1996.

[68] R. H. Økland. Rescaling of ecological gradients. II. The effect of scale on symmetry of species response curves. *Nordic Journal of Botany*, 6:671–677, 1986.

[69] R. H. Økland. Vegetation ecology: theory, methods and applications with reference to fennoscandia. *Sommerfeltia*, 1(Suppl.):1–172, 1990.

[70] R. H. Økland and O. Eilertsen. Canonical correspondence analysis with variation partitioning: Some comments and an application. *Journal of Vegetation Science*, 5:117–126, 1994.

[71] J. Oksanen. Problems of joint display of species and site scores in correspondence analysis. *Vegetatio*, 72:51–7, 1987.

[72] J. Oksanen. A note on the occasional instability of detrending in correspondence analysis. *Vegetatio*, 74:29–32, 1988.

[73] J. Oksanen. Neighbour diversity of plants. *Journal of Vegetation Science*, 8:255–258, 1997.

[74] J. Oksanen. Why the beta-function cannot be used to estimate skewness of species responses. *Journal of Vegetation Science*, 8:147–152, 1997.

[75] J. Oksanen, E. Läärä, P. Huttunen, and J. Meriläinen. Estimation of pH optima and tolerances of diatoms in lake sediments by the methods of weighted averaging, least squares and maximum likelihood, and their use for the prediction of lake acidity. *Journal of Paleolimnology*, 1:39–49, 1988.

[76] J. Oksanen, E. Läärä, P. Huttunen, and J. Meriläinen. Maximum likelihood prediction of lake acidity based on sedimented diatoms. *Journal of Vegetation Science*, 1:49–59, 1990.

[77] J. Oksanen, E. Läärä, K. Tolonen, and B. G. Warner. Confidence intervals for the optimum in the Gaussian response function. *Ecology*, 82:1191–1197, 2001.

[78] J. Oksanen and P. R. Minchin. Instability of ordination results under changes in input data order: explanations and remedies. *Journal of Vegetation Science*, 8:447–54, 1997.

[79] J. Oksanen and P. R. Minchin. Continuum theory revisited: what shape are species responses along ecological gradients? *Ecological Modelling*, 151:119–129, 2002.

[80] J. Oksanen and T. Tonteri. Rate of compositional turnover along gradients and total gradient length. *Journal of Vegetation Science*, 6:815–24, 1995.

[81] M. W. Palmer. Putting things in even better order: The advantages of canonical correspondence analysis. *Ecology*, 74:2215–30, 1993.

[82] M. Pärtel, M. Zobel, K. Zobel, and E. van der Maarel. The species pool and its relation to species richness: evidence from Estonian plant communities. *Oikos*, 75:111–117, 1996.

[83] S. Persson. Ecological indicator values as an aid in the interpretation of ordination diagrams. *Journal of Ecology*, 69:71–84, 1981.

[84] F. W. Preston. The canonical distribution of commonness and rarity: part I. *Ecology*, 43:185–215, 1962.

[85] R. D. Routledge. On Whittaker's components of diversity. *Ecology*, 58:1120–7, 1977.

[86] B. Smith and J. B. Wilson. A consumer's guide to evenness indices. *Oikos*, 76:70–82, 1996.

[87] C. J. F. ter Braak. Correspondence analysis of incidence and abundance data: properties in terms of a unimodal response model. *Biometrics*, 41:859–873, 1985.

[88] C. J. F. ter Braak. Canonical correspondence analysis: a new eigenvector technique for multivariate direct gradient analysis. *Ecology*, 67:1167–1179, 1986.

[89] C. J. F. ter Braak. The analysis of vegetation-environment relationships by canonical correspondence analysis. *Vegetatio*, 69:69–77, 1987.

[90] C. J. F. ter Braak. CANOCO - *a* FORTRAN *program for* cano*nical* c*ommunity* o*rdination by [partial] [detrended] [canonical] correspondence analysis, principal components analysis and redundancy analysis (version 2.1)*. TNO, Wageningen, 1987.

[91] C. J. F. ter Braak and L. G. Barendregt. Weighted averaging of species indicator values: its efficiency in environmental calibration. *Mathematical Biosciences*, 78:57–72, 1986.

[92] C. J. F. ter Braak and N. J. M. Gremmen. Ecological amplitudes of plant species and the internal consistency of Ellenberg's indicator values for moisture. *Vegetatio*, 69:79–87, 1987.

[93] C. J. F. ter Braak and C. W. N. Looman. Weighted averaging, logistic regression and the Gaussian response model. *Vegetatio*, 65:3–11, 1986.

[94] C. J. F. ter Braak and I. C. Prentice. A theory of gradient analysis. *Advances in Ecological Research*, 18:271–371, 1988.

[95] C. J. F. ter Braak and H. van Dam. Inferring pH from diatoms: a comparison of old and new calibration methods. *Hydrobiologia*, 178:209–233, 1989.

[96] M. Vellend. Do commonly used indices of β–diversity measure species turnover? *Journal of Vegetation Science*, 12:545–552, 2001.

[97] W. M. Venables and B. D. Ripley. *Modern applied statistics with S-plus*. Springer Verlag, Heidelberg, 3rd edition, 1999.

[98] D. Wartenberg, S. Ferson, and F. J. Rohlf. Putting things in order: a critique of detrended correspondence analysis. *American Naturalist*, 129:434–48, 1987.

[99] R. H. Whittaker. Gradient analysis of vegetation. *Biological Reviews*, 42:207–64, 1967.

[100] R. H. Whittaker. Evolution and measurement of species diversity. *Taxon*, 21:213–51, 1972.

[101] R. H. Whittaker. Direct gradient analysis. In R. H. Whittaker, editor, *Ordination of Plant Communities*, pages 7–50. Junk, The Hague, 1978.

[102] M. H. Williamson. The ordination of incidence data. *Journal of Ecology*, 66:911–920, 1978.

[103] J. B. Wilson. Methods for fitting dominance/diversity curves. *Journal of Vegetation Science*, 2:35–46, 1991.

[104] J. B. Wilson. Would we recognize a broken-stick community if we found one? *Oikos*, 67:181–183, 1993.

[105] M. V. Wilson and C. L. Mohler. Measuring compositional change along gradients. *Vegetatio*, 54:129–41, 1983.

[106] M. V. Wilson and A. Shmida. Measuring beta diversity with presence-absence data. *Journal of Ecology*, 72:1055–64, 1984.

[107] S. N. Wood. Modelling and smoothing parameter estimation with multiple quadratic penalties. *Journal of the Royal Statistical Society, Ser. B*, 62:413–428, 2000.

[108] S. N. Wood. mgcv: GAMs and generalized ridge regression for R. *R News*, 1(2):20–25, 2001.

[109] T. W. Yee and N. D. Mitchell. Generalized additive models in plant ecology. *Journal of Vegetation Science*, 2:587–602, 1991.

[110] K. Zobel, M. Zobel, and R. K. Peet. Change in pattern diversity during secondary succession in Estonian forests. *Journal of Vegetation Science*, 4:489–498, 1993.

[111] M. Zobel. The relative role of species pools in determining plant species richness: an alternative explanation of species coexistence? *Trends in Ecology and Evolution*, 12:266–269, 1997.